REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre :………………

Série :………………………

Thèse

Présentée en vue de l'obtention du diplôme de docteur en sciences

Option : **Informatique**

Titre

Les systèmes de détection d'objets dans le cadre d'un environnement routier

Par :
M. KHEBBACHE Mohib Eddine

Soutenue le : 11 / 07 /2024

Devant le jury :

M. BENNOUI Hammadi	Professeur	Université de Biskra	Président
M. BITAM Salim	Professeur	Université de Biskra	Rapporteur
M. MELLOUK Abdelhamid	Professeur	Université Paris-Est de Créteil, France	Co-Rapporteur
M. BENHARZALLAH Saber	Professeur	Université de Batna 2	Examinateur

Année universitaire : 2023 – 2024

List of Publications

The following listed are the publications related to this thesis.

Published Journal Papers

1. *KHEBBACHE, Mohib Eddine, BITAM, Salim, et MELLOUK, Abdelhamid. A Cost-effective Deep Active Learning for Object Detection in Automated Driving Systems. International Journal of Computing and Digital Systems, 2023, vol. 14, no 1, p. 840-854.*

Acknowledgements

I would like to express my deepest gratitude and thanks to my supervisor Pr. Salim Bitam, Professor at the University of Biskra, for their invaluable guidance, encouragement, and unwavering support throughout my research, which contributed significantly to its success. I would also like to extend my heartfelt thanks to my co-supervisor, Pr. Abdelhamid Mellouk, Full Professor at the University of Paris-Est Créteil VdM (UPEC), France, for their continuous assistance and insightful suggestions for pursuing my research. Your contributions have enriched this project, and I sincerely appreciate your support.

I sincerely thank Pr. Hammadi Bennoui, Professor at the University of Biskra, for doing me the honor of being the president of my jury. I would also like to thank Pr. Saber Benharzallah, Professor at the University of Batna 2, for his interest in my work by examining this thesis and for the honor he has done me by participating in this jury.

I would like to extend my heartfelt gratitude to my colleagues at LESIA (University of Biskra, Algeria) and LISSI Research Laboratory (University of Paris Créteil, France) for all the good times and collaboration.

I also express my sincere appreciation towards my colleagues at the University of El Oued and the University of Biskra, Department of Computer Sciences for providing me with all the support.

I would also like to thank Dr. Yazid Bounab, Doctor at the University of Oulu, Pr. Abdellatif Khelil, Professor at the University of El oued, and Dr. Samir Bourekkache, Doctor at the University of Biskra, for the richness and relevance of their comments, both in form and content, which have significantly contributed to improving the manuscript.

Last but not least, I am particularly grateful to my parents, my family, my wife, my daughter and my friends for their endless help and support.

Abstract

Object detection has recently become a crucial component of safety-critical perception tasks in autonomous driving. Advancements in object detection within Automated Driving Systems (ADS) have largely been driven by the success of the Convolutional Neural Network (CNN). However, supervised passive learning of a deep object detection model is computationally costly and requires a large amount of annotated data to cover the wide diversity of objects and scenarios in vehicular environments. This presents challenges due to visual similarity and variability of objects and labeling costs. Consequently, acquiring this data is a time-consuming and expensive process, often requiring domain experts to manually annotate high-quality bounding boxes. Moreover, ensuring the functional safety of ADS requires robust object detectors, especially in critical situations encountered within this environment.

In this context, the primary challenge is how to efficiently achieve the desired performance with a small set of labeled data while carefully balancing the trade-off between cost and accuracy. To deal with these limits, this thesis proposes and designs two contributions for object detection in autonomous driving, considering their characteristics and challenges: a batch-based query strategy and a Cost-Effective Deep Batch Mode Active Learning (CEDBMAL) framework. These solutions aim to ensure a robust and high-performance CNN-based object detector with low false detection rates and reduced annotation and training costs.

Unlike the single-criterion query strategy, which may pick redundant or outlier samples that negatively affect the detector's performance, our proposed batch-based query strategy automatically selects a batch of the top-ranked samples based on uncertainty and diversity criteria. These samples are more representative and informative, leading to more efficient training of the detector. To tailor classification uncertainty heuristics for deep object detection, we propose incorporating the detection model's outputs, such as classification and regression predictions, into the uncertainty metric measurement, arguing that samples inducing uncertainty in the model are typically not outliers, but rather instances that exhibit a larger object distribution and are expected to improve its performance. To reduce the annotation burden from redundant samples, we also propose using Euclidean distance as a representativeness measure, quantifying diversity in terms of similarity.

To mitigate the limitations posed by impractical batch size settings, our second proposal (CEDBMAL) combines our proposed query with labeling time prediction to design a cost-aware batch query. Initially, a set of batches with varied sizes is selected by the proposed query. Subsequently, we propose using labeling time prediction and dynamic programming to choose the best batch size by solving a 0-1 Knapsack problem under the constraints of annotation time, dataset size, and desired performance. This iterative process leads to an adaptive selection of the most useful and diverse training samples based on the cost of labeling. As a result, it becomes possible to effectively handle variations in annotation costs and significantly reduce the training and labeling expenses, both at the individual instance and batch level.

To validate our approaches, extensive experiments were conducted on the Caltech Pedestrian dataset to fine-tune a pre-trained deep object detector (Tiny-YOLOv3) for pedestrian detection tasks. The effects of classification uncertainty, regression uncertainty, score aggregation methods, and batch size during sample selection were thoroughly investigated. The experimental results demonstrated that the uniform-cost DAL based on our proposed batch-based query strategy outperformed random sampling and transfer learning baselines. Moreover, our cost-effective DAL approach further boosted the performance compared to other baseline deep pedestrian detectors and uniform-cost DAL approaches with a specific deep pedestrian detector. Notably, our approaches enabled the development of a robust deep pedestrian detector with significantly fewer parameters, making it suitable for deployment on low-resource devices, while maintaining the detection error rate below 57%, saving up to 50% of the labeling effort, increasing the number of pedestrians detected at early cycles, and alleviating batch size dependency.

Keywords: Autonomous driving, object detection, visual similarity, deep active learning, cost-effective training, pedestrian detection.

Résumé

La détection d'objets est devenue une composante critique de la perception pour la sécurité dans la conduite autonome. Les progrès en matière de détection d'objets dans les systèmes de conduite automatisée (ADS) sont largement dus au succès des réseaux neuronaux convolutifs (CNN). Cependant, l'apprentissage supervisé passif d'un modèle profond de détection d'objets est coûteux en termes de calcul et nécessite de grandes quantités de données annotées pour couvrir les divers objets et scénarios présents dans les environnements véhiculaires. Cela pose des problèmes en raison de la similarité et de la variabilité visuelles des objets ainsi que du coût de l'étiquetage. Par conséquent, l'acquisition de ces données est un processus long et coûteux, nécessitant souvent l'intervention d'experts pour annoter manuellement des boîtes englobantes de haute qualité. En outre, pour garantir la sécurité fonctionnelle des ADS, des détecteurs d'objets robustes sont indispensables, en particulier dans les situations critiques rencontrées dans ces environnements.

Dans ce contexte, le principal défi consiste à atteindre la performance souhaitée en utilisant efficacement un petit ensemble de données étiquetées, tout en équilibrant soigneusement le compromis entre le coût et la précision. Pour répondre à ces enjeux, cette thèse propose et conçoit deux contributions pour la détection d'objets dans la conduite autonome, en tenant compte de leurs caractéristiques et défis : une stratégie de sélection par lots et un cadre d'apprentissage actif profond à coût-effectif (CEDBMAL). Ces solutions visent à concevoir un détecteur d'objets basé sur des CNNs robustes, tout en réduisant les coûts liés à leur développement et à leur déploiement.

Contrairement à la stratégie de sélection à critère unique, qui peut sélectionner un échantillon redondant ou aberrant susceptible d'affecter les performances du détecteur, notre stratégie de sélection par lots choisit automatiquement un lot d'échantillons les mieux classés sur la base de critères d'incertitude et de diversité. Ces échantillons sont plus représentatifs, plus informatifs et permettent d'entraîner plus efficacement le détecteur. Pour adapter l'heuristique d'incertitude de la classification à la détection d'objets, nous proposons d'incorporer les prédictions de classification et de régression dans la mesure de l'incertitude, en soutenant que les échantillons induisant de l'incertitude dans le modèle ne sont pas des valeurs aberrantes, mais plutôt des instances présentant une distribution d'objets plus large, susceptibles d'améliorer ses performances. Pour éviter la charge d'annotation liée aux échantillons redondants, nous proposons également l'utilisation de la distance euclidienne comme mesure de représentativité, quantifiant la diversité entre instances en termes de similarité.

Afin de pallier les limites liées à la taille des lots, la deuxième proposition (CEDBMAL) combine notre stratégie proposée avec la prédiction du temps d'étiquetage pour concevoir une stratégie consciente des coûts. Initialement, un ensemble de lots de tailles variées est sélectionné par la stratégie proposée. Ensuite, nous proposons d'utiliser la prédiction du temps d'étiquetage et la programmation dynamique pour résoudre le problème de sélection du lot de taille optimale, comme un problème de sac à dos 0-1, sous les contraintes de temps d'annotation, de la taille de l'ensemble de données et des performances souhaitées. Ce processus itératif permet une sélection adaptative des échantillons d'entraînement les plus utiles et diversifiés en fonction du coût de l'étiquetage. En conséquence, il devient possible de gérer efficacement la variation des coûts d'annotation et de réduire de manière significative les coûts liés aux processus d'entraînement et d'étiquetage, tant au niveau de l'instance individuelle que du lot.

Pour valider nos approches, des expériences approfondies ont été menées sur l'ensemble de données Caltech Pedestrian afin de fine-tuner un détecteur d'objets profond pré-entraîné (Tiny-YOLOv3) pour la tâche de détection des piétons. Les effets de l'incertitude de la classification, de l'incertitude de la régression, des méthodes d'agrégation des scores et de la taille du lot lors de la sélection des échantillons ont également été étudiés. Les résultats expérimentaux ont montré que l'apprentissage actif profond à coût uniforme, basé sur notre stratégie de sélection par lots, surpasse l'échantillonnage aléatoire et les approches de transfert d'apprentissage. En outre, notre approche DAL à coût effectif a amélioré les performances par rapport à d'autres détecteurs de piétons profonds de référence et aux approches DAL à coût uniforme utilisant un détecteur de piétons profond spécifique. En particulier, nos approches ont permis le développement d'un détecteur de piétons robuste avec beaucoup moins de paramètres, adapté au déploiement sur des appareils à faibles ressources, tout en maintenant un taux d'erreur de détection en dessous de 57 %, économisant jusqu'à 50 % de l'effort d'étiquetage, augmentant le nombre de piétons détectés dans les premiers cycles et atténuant la dépendance à la taille du lot.

Mots Clés: Conduite autonome, détection d'objets, similarité visuelle, apprentissage actif profond, entraînement à coût-efficace, détection de piétons.

الملخص

أصبح التعرف على الأشياء مؤخرًا عنصرًا حاسمًا في مهمة إدراك الحرجة الخاصة بالسلامة في القيادة الذاتية. إن التقدم في ادماج هذا المجال داخل أنظمة القيادة الآلية (ADS) كان مدفوعًا إلى حد كبير بنجاح الشبكة العصبية التلافيفية (CNN). ومع ذلك، فإن تدريب نموذج عميق للتعرف على الأشياء باستخدام أساليب التقليدية للتعلم السلبي الخاضعة للإشراف يعد مكلفًا من الناحية الحسابية ويتطلب تسمية كميات كبيرة من البيانات لتغطية الكائنات والسيناريوهات المتنوعة الموجودة في بيئات المركبات. وهذا يشكل تحديات بسبب التشابه البصري وتنوع الكائنات وتكلفةً وضع العلامات. وبالتالي، فإن الحصول على هذه البيانات هو عملية تستغرق وقتًا طويلاً ومكلفة، وغالبًا ما تتطلب من خبراء المجال وضع تعليقات توضيحية على المربعات المحيطة عالية الجودة يدويًا. علاوة على ذلك، يتطلب ضمان السلامة الوظيفية لأنظمة القيادة الآلية وجود نماذج قوية للتعرف على الأشياء، خاصة في المواقف الحرجة التي تواجهها هذه البيئة.

وفي هذا السياق، يتمثل التحدي الأساسي في كيفية تحقيق الأداء المطلوب باستخدام مجموعة صغيرة من البيانات المصنفة بكفاءة، مع الموازنة بعناية بين التكلفة والدقة. للتعامل مع هذه الحدود، نقترح في هذه الأطروحة حلولاً مجدية للكشف عن الأشياء في القيادة الذاتية. على وجه التحديد، وبالنظر إلى خصائصها وتحدياتها، تم اقتراح وتصميم مساهمتين هما: استراتيجية اختيار دفعات من البيانات وإطار عمل للتعلم النشط العميق القائم على الدُفعات (CEDBMAL) فعال من حيث التكلفة. تهدف هذه الحلول إلى ضمان تصميم نموذج للتعرف على الكائنات يعتمد على شبكة CNN قوي وعالي الأداء من حيث تقليل الاخطاء اثناء عملية التعرف مع ضمان ايضا خفض تكاليف التسمية والتدريب.

على عكس استراتيجية الاختيار ذات المعيار الواحد، والتي قد تختار عينة زائدة عن الحاجة أو غريبة يمكن أن تؤثر على أداء نموذج المتعرف، فإن استراتيجية المقترحة تختار تلقائيًا مجموعة من العينات ذات التصنيف الأعلى بناءً على معايير عدم اليقين والتنوع. تعتبر هذه العينات أكثر تمثيلاً وغنية بالمعلومات وتؤدي إلى تدريب أكثر كفاءة للنموذج. لتخصيص استدلالات عدم اليقين في التصنيف للكشف عن الكائنات العميقة، نقترح دمج تنبؤات التصنيف والانحدار في القياس مؤشر عدم اليقين، بحجة أن العينات التي تسبب عدم اليقين في النموذج لا تكون عادةً قيمًا متطرفة، بل هي حالات تظهر توزيعًا أكبر للكائنات ومن المتوقع أن تحسن أدائ النموذج. لتجنب عبء التعليقات التوضيحية من العينات الزائدة عن الحاجة، نقترح أيضًا استخدام المسافة الإقليدية

كمقياس تمثيلي، وقياس التنوع من حيث التشابه.

للتخفيف من القيود المفروضة على إعداد حجم مجموعة العينات، يجمع للتخفيف من القيود المفروضة على إعداد حجم مجموعة العينات، يدعم الاقتراح الثاني (CEDBMAL) بين الاستراتيجية المقترحة بالتنبؤ بوقت التسمية لتصميم استراتيجية مدركة للتكلفة. أولاً، يتم في البداية تحديد مجموعة من الدفعات ذات الأحجام المتنوعة من خلال الاستراتيجية المقترحة. بعد ذلك، تتم عملية اختيار المجموعة ذات الحجم المناسب لقيود وقت التعليق التوضيحي، حجم مجموعة البيانات والأداء المطلوب. تتم عملية الاختيار استناد الى حل مشكلة حقيبة من ٠ إلى ١ باستخدام التنبؤ بوقت وضع العلامات والبرمجة الديناميكية. من خلال هذه العملية التكرارية، يتم تحديد حجم المجموعة وفقاًا لتكلفة التسمية ، مما يؤدي إلى اختيار متكيف لعينات التدريب الأكثر فائدة وتنوعاًا. ونتيجة لذلك، يمكن التعامل بفعالية مع الاختلاف في تكاليف التعليقات التوضيحية وتقليل التكاليف المرتبطة بعمليات التدريب ووضع العلامات بشكل كبير، سواء على مستوى العينات او المجموعة. للتحقق من فعالية حلولنا ، تم إجراء تجاربٍ مكثفة على مجموعة بيانات (Caltech Pedestrian) لضبط نموذج التعرف المدرب مسبقاًا (Tiny-YOLOv3) لمهمة التعرف على المشاة. كما تم دراسة آثار عدم اليقين في التصنيف، عدم اليقين في الانحدار ،طرق تجميع التقيمات وحجم المجموعة أثناء اختيار العينة على ادائ النموذج. أظهرت النتائج التجريبية أن أسلوب التعلم النشط ذو التكلفة الموحدة المعتمد على إستراتيجية المقترحة يتفوق على أساليب التعلم التقليدية المعتمدة على العينات العشوائية ونقل المعارف. علاوة عليِ ذلك، فإن أسلوب التعلم النشط الفعال من حيث التكلفة الخاص بنا عزز الأداء مقارنةً بنماذج عميقة أخرى للتعرف على المشاة وأساليب التعلم النشط ذو ذات التكلفة الموحدة مع نموذج محدد للتعرف على المشاة. على وجه الخصوص، مكنت مقترحاتنا من تطوير نموج عميق قوي للتعرف على المشاة العميقة ببيانات موسومة أقل بكثير، مع الحفاظ على معدل خطأ التعرف أقل من ٧٥ بالمائة، وتوفير ما يصل إلى ٠٥ بالمائة من جهد وضع العلامات وتخفيف مشكلة تحديد حجم مجموعة العينات.

كلمات مفتاحية: ،القيادة الذاتية التعرف على الأشياء، التشابه البصري، التعلم النشط العميق، تدريب فعال من حيث التكلفة، التعرف على المشاة.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

ITS:	Intelligent Transport Systems.
VANET:	Vehicle Ad-Hoc Network .
AI:	Artificial Intelligence.
AV:	Autonomous Vehicles.
CV:	Connected Vehicles.
ADS:	Automated Driving Systems.
ADAS:	Advanced Driving Assistance Systems.
ML:	Machine Learning.
DL:	Deep Learning.
CNN:	Convolutional Neural Network.
AL:	Active Learning.
DAL:	Deep Active Learning.
TL:	Transfer Learning.
RADAR:	Radio Detection and Ranging.
LiDAR:	Light Detection and Ranging.
mAP:	mean Average Precision.

Introduction

Recently, road traffic crashes have become one of the world's largest public health and injury problems. According to a recent technical report by the National Highway Traffic Safety Administration (NHTSA), 94% of road accidents are caused by human errors [14]. Consequently, enhancing road safety has emerged as a critical concern.

Over the last ten years, the field of road safety has received serious attention, with substantial research and development efforts have been directed towards improving road safety. These efforts, driven by government agencies, the automobile industry, and academic researchers, have focused on implementing a range of measures and solutions. Two broad classes of safety approaches are identified: "active" and "passive". Passive safety, also known as "post-crash approaches," involves reactive strategies and features designed to minimize injury and damage, thereby reducing the severity of accidents when they occur. These approaches focus on protecting vehicle occupants and the lives of those involved during and after a crash, by enhancing the physical structure of the vehicle by incorporating safety features such as seat belts, airbags, anti-lock brakes, energy-absorbing materials, and electronic stability control. On the other hand, various measures target improvements in roadside infrastructure. In contrast, active safety, also referred to as "pre-crash approaches," encompasses technologies and proactive strategies designed to prevent accidents from occurring and actively assist drivers in critical situations. These approaches typically focus on designing active safety systems that continuously monitor vehicle and driver status in real-time, enabling collision prediction [15].

Intelligent Transport Systems (ITS) have long been considered a key technology for increasing safety and improving transport infrastructure in the on-road environment [15]. The rapid evolution of ITS is marked by the adoption of cutting-edge technologies and data-driven insights. The incorporation of smart sensors, the Internet of Things (IoT), big data, data fusion, computer vision, cloud computing, and artificial intelligence (AI) has led to improved ITS service levels in real-world vehicular applications. They are also playing a role in the transition from connected Vehicles (CV) to Autonomous Vehicles (AV). In conjunction with conventional and electrical vehicles, they will enable the shift to more advanced ITS, like autonomous driving and green ITS, over the next decade [16].

The Vehicle Ad-Hoc Network (VANET) is a key enabler for deploying ITS applications, communication systems, and services in vehicular network environments. By supporting and including autonomous vehicles as network elements and nodes, VANET technologies can significantly reduce vehicular accidents. This is achieved by enabling cooperation among vehicles and infrastructure to improve the range, quality, and reliability of safety-related information, thereby perceiving potentially dangerous situations [15, 16].

In light of extensive research conducted in the fields of ITS and road safety, findings have highlighted the transition from manual to autonomous driving which is highly relevant for addressing key challenges in modern transportation and aligning seamlessly with the goals of ITS. In this context, research and development in automated driving technologies, as well as regulatory reforms worldwide signify a significant step forward in the development of autonomous vehicles (automated or self-driving vehicles). An AV can either assist a human driver or independently operate the vehicle based on the levels of vehicle automation. These levels indicate the extent of involvement of human drivers and automated systems in monitoring the surrounding environment and controlling the vehicle, spanning from no automation to full automation levels. Consequently, new possibilities may arise to address economic, environmental, and safety-related concerns, where the AVs have the potential to enhance road safety by mitigating or reducing some accidents caused by human error or collisions with vulnerable road users (VRUs eg. other vehicles, pedestrians, cyclists, animals) [17, 18].

Inspired by these objectives, Automated Driving Systems (ADSs) are being developed to handle the full range of dynamic driving tasks at automation levels 3, 4, and 5 when activated. Ongoing efforts are focused on designing and testing these systems to boost economic growth, promote safety, and enhance the passenger experience, with the expectation that their annual societal benefits could reach nearly $800 billion by 2050. The first attempts focused on testing the ability of prototypes to navigate autonomously in typical urban environments that model urban scenes. This was the subject of the US Department of Defense's DARPA challenges from 2004-2007, alongside other series of challenges and competitions organized by different research groups worldwide. In the meantime, existing insights have moved beyond the prototype stage to production, identifying multifaceted implications that are likely to limit progress in introducing ADS [14, 18].

ADSs are designed either as standalone, ego-only systems or as connected multi-agent systems. Furthermore, these design philosophies are implemented through two alternative approaches: modular or end-to-end driving. The typical basic functions of a modular ADS can be summarised as perception, localization and mapping, assessment, planning and decision making, vehicle control, and human-machine interface [14]. In addition, these automated systems rely on autonomous sensor technologies, such as Radio Detection and Ranging technology (RADAR), Light Detection and Ranging (LiDAR), ultrasonic, and cameras, that can perceive and therefore reach real-time vehicle surrounding situation awareness [19].

Despite the widespread deployment and advancement of AVs, road safety remains a critical challenge. In this context, the performance of perception, planning, and control modules has a significant impact on the achievement of safe autonomous driving. Besides, from a research and development perspective, accident-based statistical analysis, in conjunction with on-road, hardware, and software testing, is regarded as a useful technique for evaluating the effectiveness of implemented safety countermeasures. Furthermore, these findings drive further research aimed at identifying potential safety risks, various types of failures or accidents, and understanding the present condition of automotive technology development. In other words, safety concerns and impacts can be studied to facilitate a successful transition from manual driving to autonomous driving, and the expected reductions in crash severity and frequency can be assessed for the gradual introduction of connected and automated vehicles alongside conventional ones [17, 20, 21].

Motivation of Object Detection in Autonomous Driving

While autonomous driving technology continues to gain significance, it is still far from being mature since it faces considerable developmental hurdles. To fill this gap, one of the biggest challenges that needs to be strongly investigated by the autonomous driving research community is object detection. At the heart of the transformation from manual to autonomous driving, object detection is a sub-field of computer vision that seems very beneficial for AV technology development from several perspectives.

In autonomous driving scenarios, AV requires navigating safely and effectively through hostile, uncertain, and rapidly changing environments, such as urban areas and roads, where multiple objects such as VRUs, traffic signs, lines, street furniture, and obstacles coexist. In such a context, it's vital to equip autonomous vehicles with powerful and sophisticated perception systems to accurately perceive and comprehend their surroundings, behave appropriately, and adapt to their environment. Object detection can fit the requirement of developing a reliable AV perception system by providing semantic comprehension of the collected data through detecting and tracking other objects and recognizing their characteristics, including size, shape, color, distance, and speed. This enables autonomous vehicles to plan their route, avoid collisions with other objects, and obey traffic laws, thus, ensuring the safety of VRUs and leading to safe and efficient autonomous driving.

Furthermore, sensing information required by perception systems is acquired and collected through an array of onboard sensors, serving as a primary source integrated into an AV. However, the emergence of sophisticated, cost-effective sensors such as automotive radars, multi-layer laser scanners, and cameras with depth and appearance capabilities has paved the way for benefiting from novel information modalities and sources.

Significant advancements in the domains of computer vision and deep learning (DL) have profoundly impacted the evolution of ADS. In particular, the advent of convolutional neural networks (CNNs) has led to the emergence of novel techniques that surpass traditional state-of-the-art approaches in areas such as object detection, whether implemented as standalone modules or within fully trainable end-to-end systems. In this context, a detailed evaluation of the AVs' object detection-related application requirements will be crucial from the software development perspective, by determining the potential safety risks concerns and developing strategies that support the smooth integration of AVs into future transportation networks. For this reason, selecting the most appropriate object detection method is crucial for guaranteeing effective performance in the specified environmental conditions.

Vehicle functions pose functional safety problems regardless of underlying malfunction and failure patterns and stages. These problems can lead to critical safety issues if not carefully controlled and maintained beforehand. In this regard, scene understanding, the most crucial step in autonomous driving, relies on detecting the surrounding objects of a vehicle. Therefore, failure to identify those objects correctly on time can cause irreparable damage. Although DL-object detection approaches aim to reduce false detection, the main drawbacks of OD-based safety-critical perception tasks are heavily dependent on deep learning specific insufficiencies and sensor failures [22, 17, 23].

The aforementioned context motivates us, in this thesis, to contribute to the research effort by exploring solutions to object detection shortcomings, with an emphasis on guaranteeing robust deep object detection algorithms necessary for the functioning of the AV's perception systems.

Problem Statement

This thesis was proposed to address the challenges of improving object detection in the autonomous driving field, focusing on the following aspects:

- **Autonomous driving datasets constraints**: Perception models based on deep learning typically require large-scale and diverse training datasets to ensure consistent performance in different driving conditions. Traditionally, these datasets are compiled by gathering extensive visual frames from urban public areas and manually annotating each object of interest within these frames. However, due to the diverse classes (background and foreground) and varying density distributions of on-road objects (sparse and dense), training samples collected from non-deterministic urban scenes often exhibit biases, data imbalances, and repeating visual patterns. Furthermore, the process of gathering and annotating such a wide array of data is costly and time-intensive. Although synthetic data generation through simulation offers a cost-effective alternative to augmenting real-world data, creating realistic synthetic data remains challenging.

- **Training efficiency**: Training DL-based object detection models for real-time identification of objects, irrespective of image distortions or meteorological conditions in real-world applications such as autonomous driving, is inherently challenging due to the constraints of the passive supervised learning paradigm. This learning scheme requires training the model using fully annotated training data. However, annotating data in large-scale autonomous driving datasets is particularly difficult due to several factors, including the labour-intensive nature and high costs of manual labeling by domain experts, challenges in obtaining representative data and accurate annotations, and issues related to noisy and less effective samples during annotation and training process. To address these challenges, a primary hypothesis proposes optimizing annotation and training resources by selectively annotating a subset of informative samples that maximize performance gains while minimizing human labeling effort, with the constraints of the annotation budget. This approach emphasizes capturing diverse visual patterns. Active learning (AL) emerges as a promising alternative learning paradigm that offers a label-efficient iterative learning scheme for addressing the limitations of supervised models. This is achieved through actively selecting a small subset of relevant data for manual annotation, thereby maximizing its impact on model training. However, the primary challenge lies in determining effective criteria for selecting the most valuable instances and deciding how many instances to query at a time, considering the inherent characteristics of DL training and dataset instances. Additionally, given the substantial manual effort required for data preparation, integrating label-efficient learning schemes into the training pipeline of object detection systems presents another significant issue.
- **Variable labeling cost**: In several autonomous driving scenarios, manual labeling by domain experts is a critical but time-consuming and cost-sensitive process with a limited annotation budget. Consequently, annotation cost is a significant issue and major limitation. Although the impact of annotation cost can be assessed, most passive learning schemes, such as supervised learning, did not consider it during training DL-based object detection models. A cost-effective approach is needed to incorporate annotation cost during training while addressing trade-offs between accuracy and cost. Some sparse batch-mode DAL solutions have been proposed for generic objects and other vision tasks. However, such approaches could not

adopted for object detection tasks, particularly in the context of on-road objects, when assuming a fixed batch size and labeling cost. In autonomous driving scenarios, these assumptions, without accounting for the correlation between the number of objects of interest and the varied labeling cost across dataset instances, may not lead to optimal generalization accuracy, effective management of labeling cost variability, practical batch size determination, or minimized redundancy.

Contributions

In this Doctorate thesis, our main contributions to overcome the aforementioned drawbacks related to the training cost and deployment of a deep object detector in autonomous driving include:

- The first contribution is an efficient and scalable batch sampling query strategy designed to deal with class imbalance, visual pattern similarity, and domain-shift during selection from a large-scale sequence of highly similar unlabeled frames. Our proposed strategy leverages the uncertainty of both classification and regression model outputs to elaborate an uncertainty-based sampling metric for informativeness measurement while employing Euclidean distance as a similarity metric for representativeness measurement. Additionally, it examines various aggregation functions to generate image-level scores for ranking purposes. This approach aims to query more informative, less noisy (outliers), and less redundant instances, primarily based on the density of the most promising predicted objects.

- The second contribution is a new training framework called Cost-Effective Deep Batch Mode Active Learning framework (CEDBMAL), featuring a label-efficient iterative learning algorithm. Our proposed CEDBMAL aims to incrementally improve the performance of a single-stage CNN-based object detector with less training and labeling costs. By enhancing the proposed query strategy, our proposed CEDBMAL leverages labeling time prediction and dynamic programming to solve the optimal batch size determination under the constraints of a given budget and annotation cost variation at batch-level. Additionally, our proposed CEDBMAL investigates the integration of other label-efficient iterative learning schemes, such as transfer learning(TL) alongside AL within the same training framework. This approach aims to acquire the labeling of a less noisy and more diverse batch of frames with size aligned with minimal annotation cost varied over the batches. At that time, the selected batch is subsequently used as a mini-batch to train an efficient and robust object detector model without extra burden.

- The third contribution is the training of a tiny version of an object detector and the estimation of the annotation time, expressed as a function of a batch rather than of a single sample, specifically in the context of autonomous driving. This approach allows us to explore the potential for building comparative high-performance deep object detectors with simpler architecture and fewer parameters, rendering them suitable for deployment on low-resource devices.

Thesis Organisation

The rest of this thesis is organized as follows:

- **Chapter 2**: Provides an overview of the thesis context, detailing the background, architectural designs, and key implications of autonomous vehicles. This chapter also highlights the significant challenges that arise and how they can be addressed in this context.
- **Chapter 3**: Introduces, reviews, and discusses object detection in the field of autonomous driving. This chapter details the fundamental concepts of object detection, as well as recent architectures, and cost reduction methods proposed in the literature. It also examines datasets and metrics and presents the existing solutions suggested for object detection in autonomous driving literature.
- **Chapter 4**: Presents an overview of active learning, with a particular focus on its applications in autonomous driving. It introduces the concepts and theory underlying Deep Active Learning (DAL)-based approaches, the diverse query strategies, scenarios, and settings that our proposed contributions are based on and inspired by. It also reviews the state-of-the-art applications of DAL for object detection in autonomous driving, discusses the weaknesses and inadequacies of such techniques, and critiques their limits, shortcomings, and challenges to propose effective solutions based on this analysis.
- **Chapter 5**: Describes the general process of designing and developing our cost-effective training framework. It details the design of our batch-mode query strategy to improve Deep Batch Mode Active Learning for object detection. It then presents the enhancement of this strategy integrated into our proposed training framework, named Cost-Effective Deep Batch Mode Active Learning framework (CEDBMAL). Finally, it illustrates and discusses both quantitative and qualitative results of performance evaluation and comparison with similar solutions.
- **Conclusion**: Summarises this thesis, outlines the findings toward object detection in autonomous driving, and suggests some future research directions in this domain.

Chapter 1

Autonomous Driving: An Overview

Introduction

Over the past decade, autonomous driving (or driving automation) has drawn significant interest from both academia and industry. With today's advancements, AVs (automated vehicles or self-driving cars) have the potential to revolutionize human mobility and provide a safer driving experience. By leveraging advanced technologies, their deployment promises substantial economic and societal benefits, including traffic congestion mitigation, decreased energy consumption, and reduced driving-related errors. From another point of view, "driving automation" encompasses both *Advanced Driver Assistance Systems* (ADAS) and *Automated Driving Systems* depending on the level of automation.

This chapter provides a comprehensive understanding of the current state of autonomous driving technologies. Section 1.1 delves into the technological background of transportation systems and presents a taxonomy of contemporary road safety approaches. Section 1.2 then examines the historical development of autonomous vehicles, classifying these systems based on various criteria, including levels of driving automation. This section also outlines the evolution of high-level system architectures and their implications and highlights emerging trends addressing current challenges. By examining the technological landscape and current issues, this chapter represents a step forward towards in-depth discussions and proposed solutions in subsequent chapters.

1.1 Intelligent Transportation Systems and Road Safety

1.1.1 Intelligent Transportation Systems

Intelligent Transportation Systems have long been recognized as pivotal technologies for enhancing safety and improving transport infrastructure in the on-road environment. ITS communication systems, which enable connected vehicles to perform vehicular communications and cooperate with each other, are an important pillar of ITS applications [15].

Today, smart sensors, IoT, big data, data fusion, computer vision, cloud computing, and AI are crucial enablers for advancing ITS service levels in future smart cities. These technologies are also pivotal in transitioning from connected vehicles to autonomous ones. Together with conventional and electrical vehicles, they will drive the evolution towards more advanced ITS, including autonomous driving and green ITS, over the next decade [16].

The Vehicle Ad-Hoc Network serves as a vital communication framework within the broader ITS infrastructure, facilitating the deployment of ITS applications and services in vehicular network environments. As a specialized form of Mobile Ad-hoc Network (MANET), VANET consists of moving vehicles acting as nodes. It is a spontaneous, self-organized, and distributed network deployed in on-road environments. Its unique characteristics include predictable mobility, lack of power constraints, high computational ability, large scale and strength with variable density, and rapid changes in network topology. Intending to deploy a cooperative driving system, VANET has emerged as an active area of research, standardization, and development. Numerous researchers and vehicle manufacturers have proposed and developed a broad range of safety and non-safety applications. These applications are expected to improve road safety, traffic efficiency, and driver and passenger comfort. They require a specialized data structure format, along with inter-vehicle (V2V) and vehicle-to-infrastructure (V2I) networks to disseminate and share different types of information. Examples include safety information for accident prevention and data about weather, traffic flow, and points of interest (gas stations, shopping malls, and fast food) to improve passenger comfort and traffic efficiency. To achieve this, VANET adopts Wi-Fi-based and cellular-based access technologies to provide wireless connectivity among connected vehicles. Additionally, significant efforts have been devoted to standardizing vehicular communication to support various applications on such vehicles.

Popular wireless technologies employed within the context of vehicular communication standards encompass IEEE 802.11p, Dedicated Short Range Spectrum (DSRC), and Wireless Access in Vehicular Environments (WAVE). Besides, other communication technologies and standards, including WiMax and Bluetooth, are also used [24]. Nowadays, the prospect of vehicles sharing and exchanging data with a range of other devices, including pedestrians' handheld computers, bicycles, ground stations (GN), and unmanned aerial vehicles (UAV), is becoming increasingly feasible as the automotive industry develops. Consequently, VANET necessitates heterogeneous cooperation with other wireless technologies and network infrastructures, achieved through vehicle-to-everything (V2X) communication [25].

Although existing standards are currently being used to deploy VANET networks, the automotive industry and research community are addressing the deficiencies and shortcomings of these standards. New standards, such as IEEE 802.11bd (for DSRC) and 5G NR V2X (for C-V2X), are being developed to meet the high reliability, low latency, and Quality of Service (QoS) requirements of future autonomous driving applications [26].

Alongside Trusted Authorities (TAs) and Roadside Units (RSUs), connected vehicles play a pivotal role in facilitating VANET deployment. Each vehicle node is equipped with an On-Board Unit (OBU), a radio interface enabling vehicular communication to support cooperative data and information exchange necessary for various vehicular applications. Furthermore, a set of in-vehicle sensors is embedded to collect and process information about the surrounding environment [24]. Recently, inter-vehicle communication systems using Wi-Fi on Android smartphones have also been explored as an alternative to the 802.11p OBU before its widespread use on modern vehicles [27].

Currently, Blockchain and AI-based VANET are emerging as promising approaches for contributing to the fulfillment of the Sustainable Development Goals (SDGs). They effectively address various issues within VANETs, including routing protocols, security and privacy, and meeting secure ITS applications requirements [28, 29]. Moreover, cost-effective communication technologies such as device-to-device (D2D) and mmWave communication, satellite communication, coupled with softwareization techniques like Software-Defined Networking (SDN), fog computing, and IoT are contributing significantly to the advancement of VANETs and the evolution of emerging Internet of Vehicles (IoV) applications [25]. Figure 1.1 summarises the relationship between the aforementioned technologies.

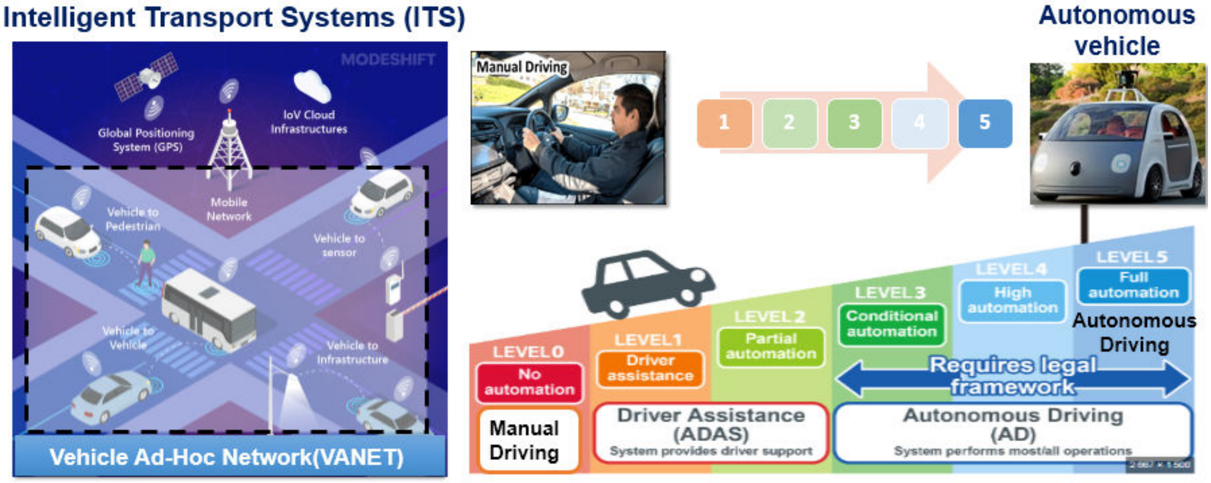


Figure 1.1: Transition from manual driving to autonomous driving.

Despite the extensive implementation and advancement of ITS, road safety remains a critical challenge. Achieving substantial ITS deployment will likely require significant efforts to address the various issues associated with this challenge. In light of the above, the next subsection examines details regarding the concept of road safety.

1.1.2 Road Safety

Over the past decade, safety on roads, or the “road safety problem,” has received serious attention. To address this issue globally, a range of measures and solutions have been implemented by government agencies, the majority of automobile manufacturers, and academic researchers. Figure 1.2 presents a taxonomy of various approaches and strategies for improving road safety.

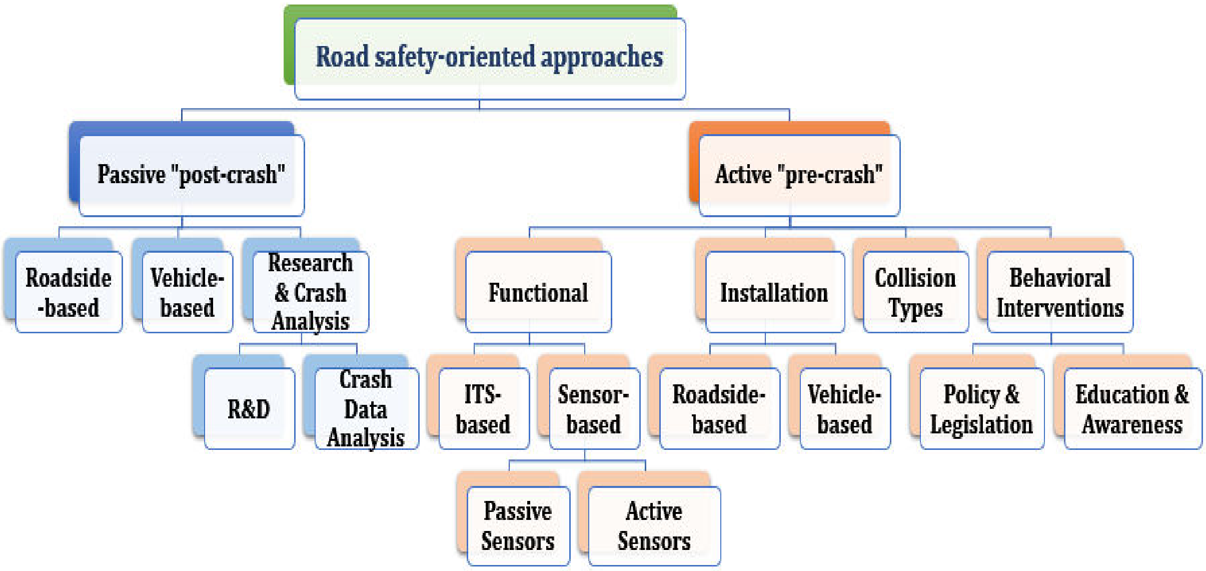


Figure 1.2: Taxonomy of road safety-oriented approaches.

As shown in Figure 1.2, this taxonomy differentiates between "active" and "passive" safety approaches. Passive safety (or "post-crash" safety) approaches encompass reactive strategies and features oriented towards minimizing injury and damage, thereby reducing the severity of accidents when they occur. Consequently, vehicle occupants can be protected, and lives can be preserved during and after a crash.

A multitude of measures within this category reinforce the vehicle’s physical structure by incorporating safety features such as seat belts, airbags, anti-lock brakes, energy-absorbing materials, and electronic stability control. Nevertheless, other measures focus on enhancing roadside infrastructure, including [15, 30]:

- Enhancing road geometry, signage, and visibility to limit accident-prone areas when designing roads.
- Optimizing traffic flow, controlling speed limits, and effectively using traffic signals for traffic management.

- Installing physical barriers to prevent vehicle collisions with vulnerable road users such as pedestrians or animals, as well as with other on-road objects.

Conversely, active safety (or "pre-crash" safety) approaches typically focus on designing active safety systems that provide real-time vehicle and driver monitoring for collision prediction. This proactive approach actively assists the driver in preventing accidents or minimizing their impact in critical situations. The primary objective is to enhance the driver's/user's awareness of potential collision risks through warnings, assistance, behavior modification, or partial automation of the driving process. For instance, well-studied active safety systems include collision warning, avoidance, and mitigation systems, as well as object detection systems [15].

The effectiveness of an active safety system can vary depending on factors such as collision type, installation location, visibility conditions, and available surrounding information. In scenarios of limited vehicle visibility, ADAS relies on in-vehicle sensors to capture and collect ambient environment data. Interpretation of the sensor signals enables continuous monitoring of the vehicle and driver state. Commonly sensors used for this purpose include passive sensors such as laser, ultrasound, infrared, and radar, as well as active sensors like cameras [15, 31]. Additionally, smartphones can function as front vision sensors, placed between the rear-view mirror and the windscreen [32].

Effective cooperation among active safety systems has the potential to improve safety in conditions of low visibility. This cooperation may entail assistance and support from the road infrastructure, other vehicles, or both. ITS-based approaches, leveraging VANET communication techniques, have exhibited promising advantages for cooperative active safety systems. Within a vehicle network, safety-related information can be efficiently disseminated through V2V and V2I communication. This information, including sensor data from both infrastructure-based and in-vehicle sensors, or urgent messages, plays a pivotal role in alerting the assisted driver and/or influencing their maneuvers [15, 33]. Moreover, leveraging the roadside active safety system, RSU-based cooperative perception is tailored to effectively support V2X communication as an integral component of cooperative ITS [34].

From a research and development perspective, utilizing accident-based statistical analysis coupled with on-road hardware and software testing is valuable for evaluating the effectiveness of implemented safety countermeasures and the expected reductions in crash severity. The findings of this evaluation drive ongoing research to identify safety risks, understand various types of failures or accidents, and advance our knowledge of the current state of automotive technology development. Additionally, studying safety concerns and their implications facilitates a smooth transition from manual to autonomous driving, assessing the gradual integration of connected and automated vehicles alongside conventional ones [17, 20, 21].

Based on active and passive strategies, a global approach to road safety has been introduced to create effective road safety programs. For example, the United Nations General Assembly (UNGA) adopted a resolution establishing the "Decade of Action for Road Safety" (2011-2020 and 2021-2030). Additionally, several countries have planned effective road safety programs that include legislative changes and road safety laws (such as seat belt use, speed reduction, and banning the use of mobile phones during driving) to improve the behavior of road users.

Given in-depth ITS and road safety research, findings reveal that autonomous driving is highly relevant to addressing modern transportation challenges and aligns perfectly with ITS objectives. However, an overview highlighting the transition from manual to autonomous driving is essential, focusing on vehicle automation levels, the current state of AV development and deployment, and technological and scientific bottlenecks. This provides the fundamental knowledge for further improving road safety, as discussed in the next section.

1.2 Autonomous Driving: A New Era of ITS

The rapid evolution of ITS, marked by the incorporation of cutting-edge technologies and the emergence of technology-driven road safety measures, has facilitated the seamless transition toward autonomous driving. The integration of AVs, CVs, and electrical vehicles alongside conventional vehicles, coupled with the design of systems aligned with ITS objectives in terms of safety and efficiency, has demonstrated the potential to fulfill the promise of safe, cost-effective, intelligent, and sustainable ITS.

This section presents an overview of the fundamental principles underlying autonomous driving, encompassing architectural considerations, development processes, safety implications, and associated challenges.

1.2.1 From the Lab to Real-world Application Use

Driven by convenience, safety, and economic benefits, autonomous driving and autonomous/self-driving vehicles (cars) have become subjects of intensive study from both development and real-world testing perspectives. Several key areas related to the performance of functional automated driving tasks are explored in detail below.

- **AV Projects and Challenges**: The timeline of AVs began in 1980 with the Eureka Project, known as the "Programme for European Traffic of Highest Efficiency and Unprecedented Safety (PROMETHEUS)," marking the initial milestone in the development and testing of automated driving. From 1987 to 1995, Daimler-Benz developed, through this project, a vision-guided robotic vehicle called VITAII and tested its ability to drive autonomously on European motorways [35, 36].

In contrast to earlier efforts, recent advancements include gradual deployments of new semi-automatic and fully automatic driving features in vehicles. These innovations leverage cutting-edge technologies, including artificial intelligence, machine learning, and advanced sensors, driving the 21st-century autonomous vehicle revolution.

Initial efforts focused on testing prototypes' ability to navigate autonomously in typical urban environments. This was the subject of the US Department of Defense's DARPA challenges (2004-2007) and other series of global competitions [14]. Since 2009, a notable shift from development and testing to real-world deployment and production has occurred. Currently, major automotive manufacturers (Ford, Mercedes Benz, Volkswagen, Audi, and BMW) and IT companies (Google, Uber, NVIDIA, and Tesla) are developing their autonomous vehicles. For example, Google's self-driving car, known as 'Waymo,' has undergone real-world testing in complex urban traffic scenarios [37].

In the coming decades, advancements in development tools, algorithms, and processing hardware are expected to lead to significant progress in the large-scale deployment of autonomous vehicles, ultimately reaching a high automation level. Autonomous vehicles are already being tested on public roads in the US, Europe, and East Asia. The National Highway Traffic Safety Administration (NHTSA) has provided estimates for commercial availability and deployment timelines, as illustrated in Figure 1.3 [38].

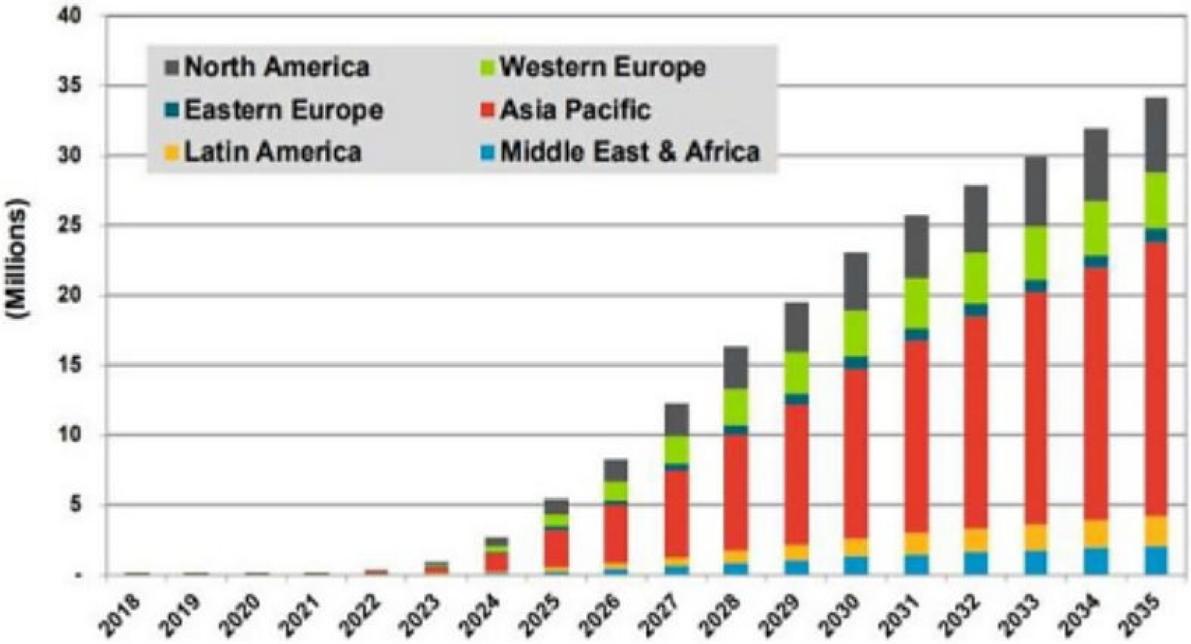


Figure 1.3: Estimated AV Deployments by Region.

Despite progress and research achievements, autonomous vehicle benefits, costs, impacts, and implications are still under investigation. Extensive literature attempts to predict these factors using various approaches [39]. These investigations have led to the classification of automation levels and define of current capabilities, technologies, and functionalities at each level, as described in The following subsection.

1.2.2 Autonomy Level Evolution

The transition between different stages of autonomous vehicle development is closely related to balancing autonomous and manual control dimensions, requiring a standardized classification of automation levels. Two primary approaches have been proposed in the literature.

The first approach, proposed by Flemisch et al. [40], focuses on vehicle transitions from manual to fully automated control. Accordingly, the envisaged classification spans from providing driver assistance and warnings to achieving complete automation of the driving task. Figure 1.4 illustrates the transitions between different levels of assistance and automation [15].

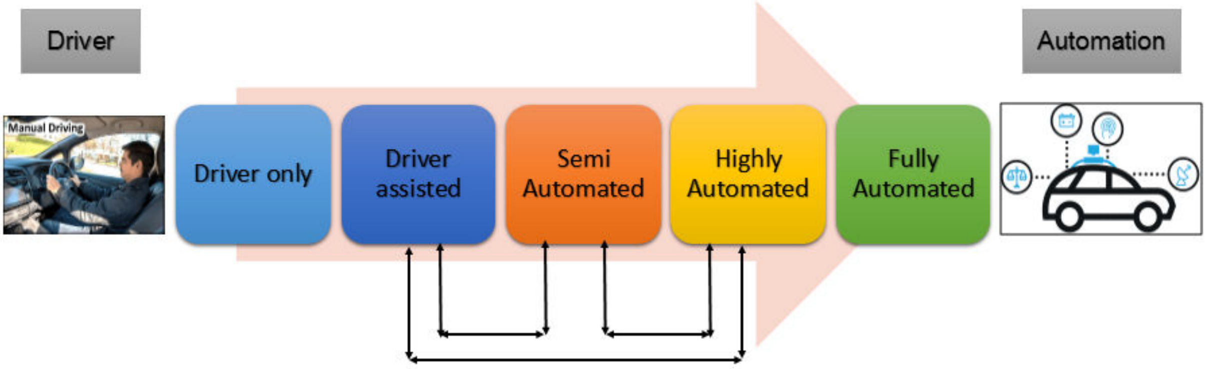


Figure 1.4: Transition between different levels of assistance and automation.

As shown in Figure 1.4, the vehicle system may transition from driver-only to driver-assisted, semi-automated, highly automated, and fully automated states.

Alternatively, the second approach focuses on establishing a standard, as exemplified by the Society of Automotive Engineers (SAE) case. Figure 1.5 illustrates the SAE J3016 standard, which defines a 0-5 scale for categorizing autonomous driving capabilities into different levels [41].

As illustrated in Figure 1.5, semi-automatic levels (L0-L2) pertain to advanced driver assistance systems providing basic driver assistance, with the driver monitoring and controlling the vehicle. Conversely, higher levels (L3-L5) progress beyond ADAS, where automated driving systems actively take almost total control of driving tasks without human intervention. Level 5 represents the anticipated future of highly automated vehicles.

Although technological boundaries for each level are well-defined, ongoing research focuses on design and development implications regarding potential bottlenecks, constraints, trade-offs, and performance-related requirements. Subsequent subsections detail these critical implications.

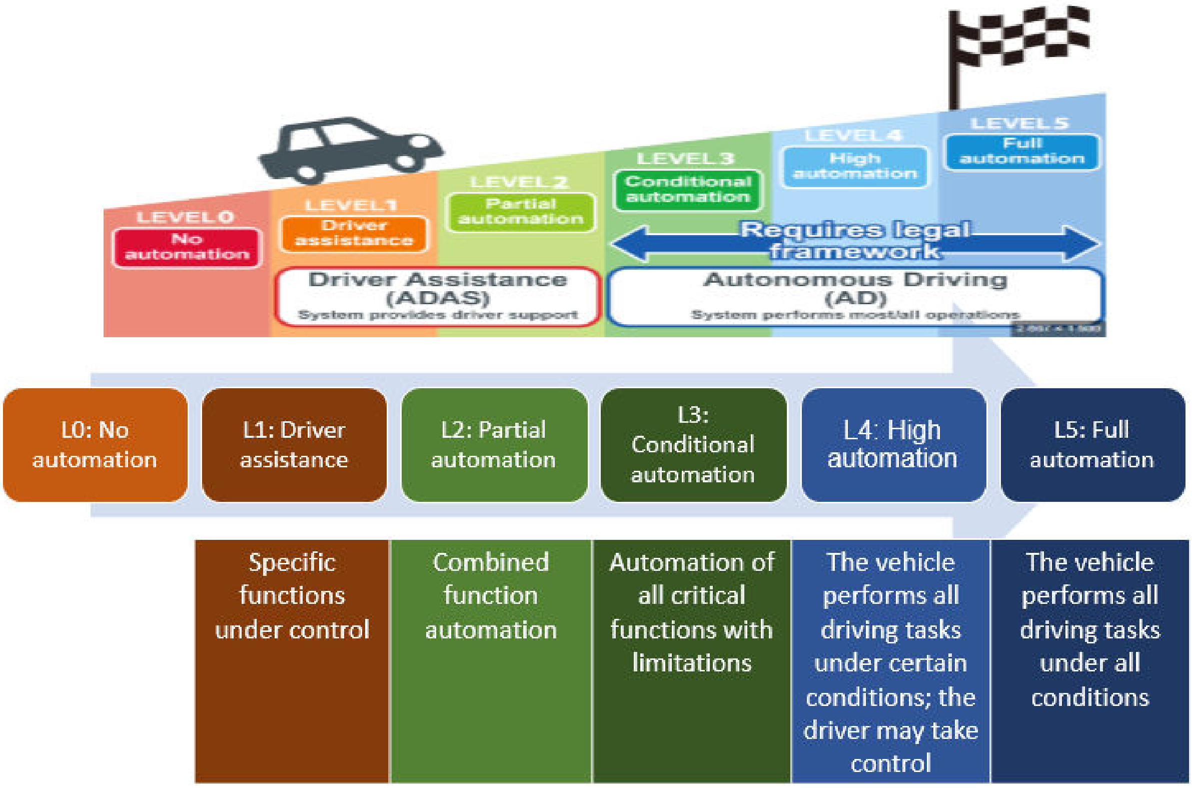

LEVEL 0
No automation
LEVEL 1
Driver assistance
LEVEL 2
Partial automation
LEVEL 3
Conditional automation
LEVEL 4
High automation
LEVEL 5
Full automation
Requires legal framework
Driver Assistance (ADAS)
System provides driver support
Autonomous Driving (AD)
System performs most/all operations
L0: No automation
L1: Driver assistance
L2: Partial automation
L3: Conditional automation
L4: High automation
L5: Full automation
Specific functions under control
Combined function automation
Automation of all critical functions with limitations
The vehicle performs all driving tasks under certain conditions; the driver may take control
The vehicle performs all driving tasks under all conditions

Figure 1.5: Levels of autonomy within the scope of SAE J3016 standard.

1.2.3 Architectural and Development Implications

Autonomous driving systems can be broadly classified as either ADAS or ADS, each with distinctive functionalities and constraints. Consequently, their design must satisfy rigorous performance standards in terms of efficiency, real-time processing, and operational and functional safety. Identifying these key design and development considerations is crucial.

- Advanced Driver Assistance Systems:

Falling within the semi-autonomous driving category (L0-L2), ADAS is an active safety system that intends to enhance driver situational awareness and safety by providing critical information, automating complex or repetitive tasks, and mitigating the severity of accidents. ADAS can also manage vehicle actuators in fully autonomous driving. Since 2000, ADAS has seen consistent growth driven by technological advancements [42]. To address driver assistance issues, industry reports indicate that collaborative efforts have been made in several projects such as CARSENSE (2000–2002) [43], INVENT (2001–2005), and PREVENT (2004–2008) [44]. By 2015, around 50-60% of vehicles were equipped with driver assistance systems, representing a market value of 7 trillion USD [31]. ADAS applications in safety-related disciplines vary depending on sensor types and placements, as seen in Figure 1.6 [42].

As illustrated in Figure 1.6, ADAS, such as object detection systems, are integrated

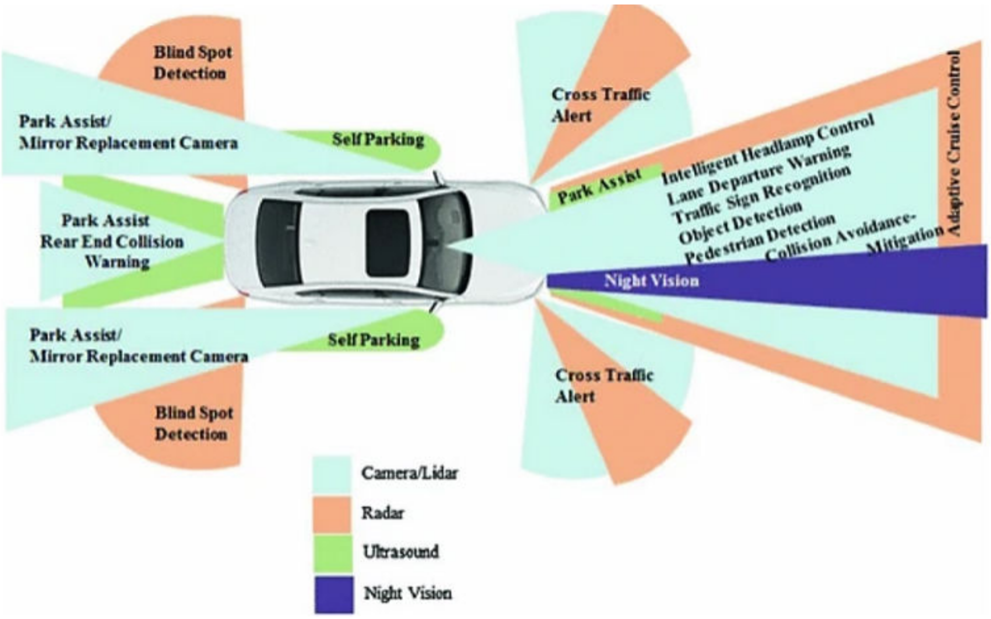


Figure 1.6: Key application areas for Advanced Driver Assistance (ADAS) using different colors to indicate type and position of Sensor [42].

into vehicles as critical embedded systems characterized by closely integrated hardware and software components. Figure 1.7 depicts an overview of the main components of an ADAS architecture.

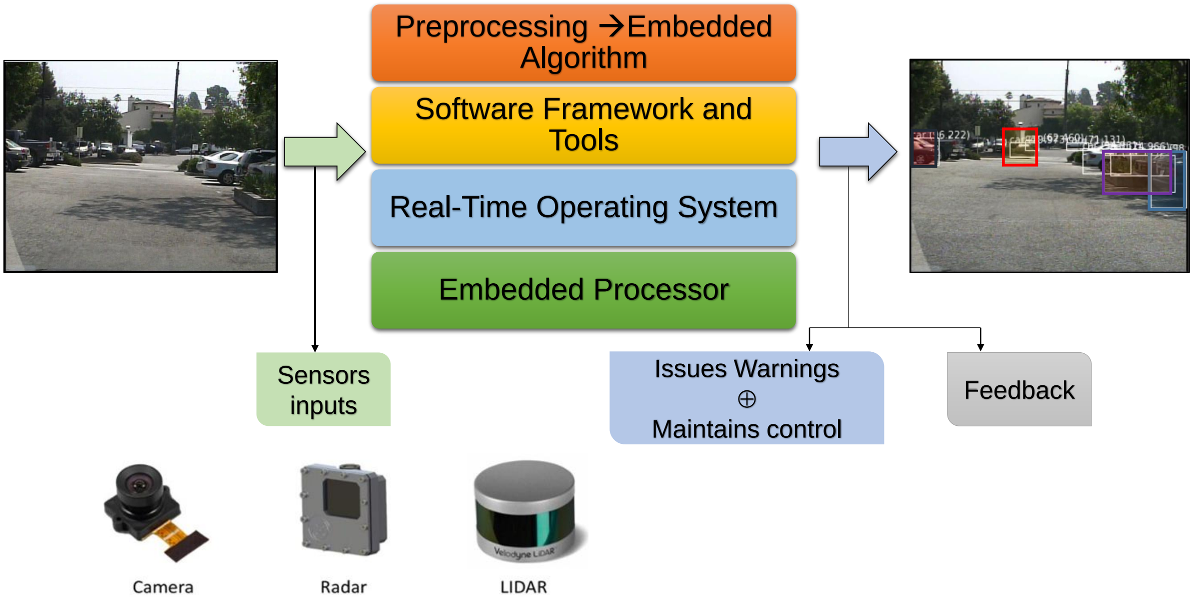


Figure 1.7: Primary building blocks of an ADAS architecture.

As illustrated in Figure 1.7, the system engine maps input sensor data, acquired using various in-vehicle sensors, into appropriate actions by processing them through a decision mechanism. These actions notify the driver and/or maintain control over vehicle functions. Specifically, input sensor data is pre-processed and transformed into a format suitable for ADAS function-related embedded algorithms (e.g., embedded vision algorithms), serving as the ADAS cognitive core. These embedded algorithms require dedicated execution environments tailored to their functions, featuring tools and a software framework that maps processes to tasks through real-time operating systems (RTOS). The RTOS in ADAS manages a range of tasks, including general operating system tasks and ADAS-specific ones like security mechanisms [42].

From a technological perspective, achievements in fields like artificial intelligence have significantly benefited the development of embedded algorithms. For instance, ML and DL algorithms build models for interpreting vision sensor data (cameras, image sensors), enabling vision-based ADAS systems to effectively detect a wide range of objects on the road, estimate their trajectories and intentions, and ultimately improve road safety [45]. For more details, please refer to [46].

For guaranteeing consistent and reliable functioning, ADAS development, implementation, and verification phases should comply with SPICE (ISO/IEC 15504) and the functional safety standard (ISO 26262) [47]. Furthermore, the development pipeline should address the key challenges associated with the implications listed below:

- **Robust performance**: ADAS should operate effectively in diverse weather and lighting conditions, with occlusion, or sensor failures. In case of deactivation, the system should automatically notify the driver of its non-operational status.

- **Accuracy**: The system should maintain high-precision sensor data processing and decision-making to avoid false positives/negatives and ensure safety.

- **Real-time processing**: The system should process vast sensor data and make timely decisions necessary for safe vehicle operation. This requires highly efficient algorithms, powerful processing hardware, and a reduction in the amount of information the driver should react to.

- **Efficiency**: ADAS should optimize algorithms, software, and systems to meet resource and time constraints imposed by the operating system.

- **Lack of industry standards**: ADAS should be able to cope with the lack of industry standards for embedded processing algorithms and variability in provided functions, data types, and sensor settings.

- **Safety, security, and reliability**: Emphasis on safety, security, and reliability in all ADAS development aspects.

- **Testing and validation**: The system should address extensive testing and validation challenges to achieve required performance standards under all possible scenarios.

Moreover, the awareness of the feedback provided by these systems has led to the emergence of innovative user interfaces (UIs), including speech dialog systems and LED patterns, enhancing the quality of infotainment experiences. Even so, designing and testing user-friendly systems remains challenging due to the diverse user skills, technical backgrounds, and qualifications of the intended users [48].

- Automated Driving Systems:

With the emergence of ADS (L3-L5), the prospect of deploying fully automated vehicles in upcoming years has become a reality. To this end, the NHTSA has provided guidance and technical assistance for AV-related research and development, extending from AV 2.0 to AV 4.0. Unlike ADAS, which assists the driver, ADS can control the entire driving task. Consequently, ADS architecture shares a common pipeline mapping sensory inputs to actuator outputs, addressing specific road environmental requirements, trade-offs, and safety and design considerations [49]. Literature reviews classify ADS architectures from different perspectives, as described below [14, 50, 51, 52, 53]:

- **Functional architecture vs. Software architecture**
  - *Functional architecture*: This approach systematically develops and manages ADS functionalities while maintaining compliance with safety standards. It outlines how ADS architecture is decomposed into high-level functional blocks, as defined by the ISO 26262 functional safety standard. It focuses on functional flows between blocks and their distribution within the ADS.
  - *Software architecture*: This classification examines the hierarchical organization of ADS infrastructure components, categorized as high-level software or low-level hardware. This structure defines each component's role clearly, facilitating a systematic flow of information and control from sensors to actuators.
- **End-to-end learning vs. Modular approaches**
  - *Modular approaches*: Modular architectures divide the ADS into distinct modules, organized as a pipeline connecting sensory inputs to actuator outputs. Each module carries out particular functions, including perception, planning, and control, which cooperate to achieve autonomous driving, like ego-motion generation. Figure 1.8 presents a typical modular ADS architecture.
  - *End-to-end learning*: This approach avoids error propagation and minimizes handcrafted components in modular architecture by generating ego-motion directly from sensory inputs using a single neural network. Training this model end-to-end maps raw sensor inputs directly to driving actions.
- **Ego-only vs. Connected systems**

- *Ego-only systems*: This viewpoint emphasizes on-board sensors and embedded processing built into a standalone platform to perform automatic driving activities. This centralized approach focuses on perceiving and reacting to the vehicle's immediate surroundings without extensive communication or data sharing with external sources or other vehicles.
- *Connected systems*: This perspective addresses ADS communication needs with external entities. This distributed approach is oriented toward enhancing situational awareness and decision-making through communication with other vehicles and infrastructure components.

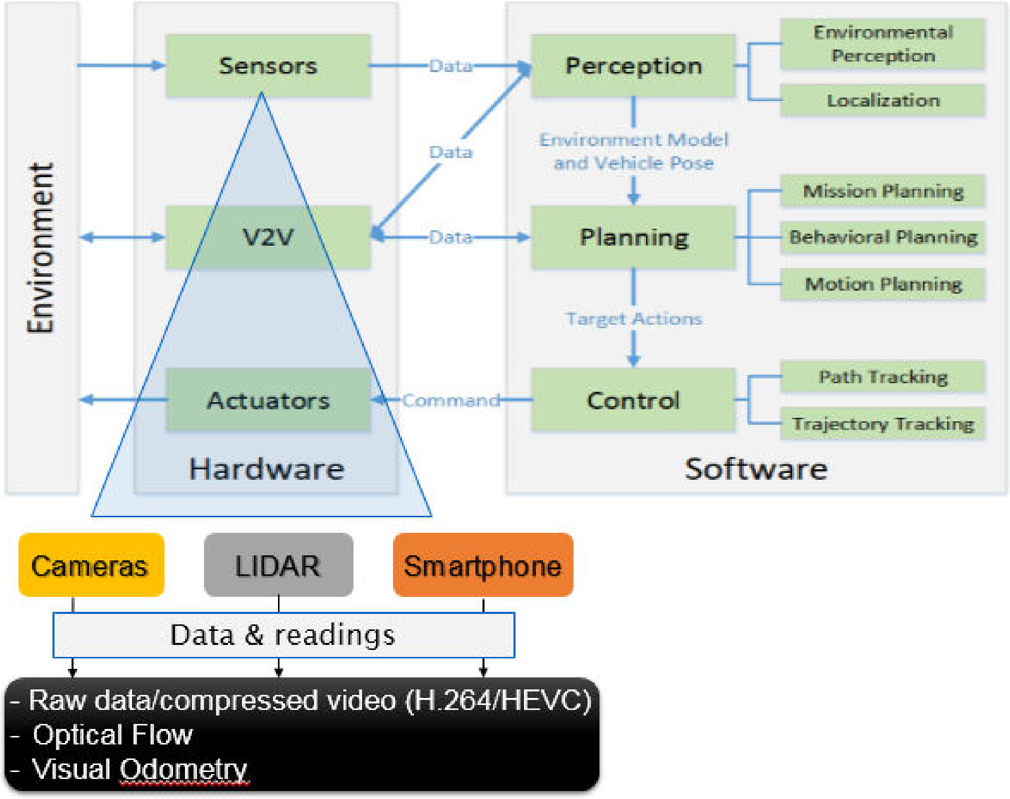


Figure 1.8: Primary building blocks of an ADS architecture.

Alternative classifications introduce entirely new requirements for the VA's Operational Design Domain (ODD) to specify precise conditions under which the system can operate. This includes considerations related to infrastructure data processing sites (Cloud-based architecture) [54, 55, 56], Human-Machine Interface (HMI) [57], safety assurance [58, 59], and other relevant factors.

Recently, the functional architecture has been extensively studied in the literature [53, 60]. As outlined by [53], it is constructed "on a software stack" comprising input/output devices, an operating system, autonomous driving frameworks, and middleware. The specific task execution of each functional component and their interrelations are presented in a modular pipeline as follows [60]:

- **Perception and localization**: The perception component is crucial for understanding the surrounding driving environment by collecting and interpreting surrounding information. It uses proprioceptive sensors for monitoring the vehicle's internal state and exteroceptive sensors, such as cameras, LiDAR, RADAR, and ultrasonic sensors, for scene interpretation and object detection. In addition to these sensors, cooperative perception is facilitated through Vehicle-to-Everything (V2X) communication, which enhances situational awareness by sharing information with external entities. As a sub-functional component, localization involves determining vehicle position relative to identified objects using the Global Positioning System (GPS) and Inertial Measurement Unit (IMU) data. To improve measurement accuracy and robustness against noise, clutter, and adverse weather conditions, sensor fusion modules are often employed to combine sensor data. Additionally, to enhance the driving experience, the user interface module is used.
- **Decision and planning:**: The primary responsibility of this stage is to intelligently determine a collision-free trajectory for safe AV navigation. In contrast to rule-based methods, a learning-based approach uses data from the perception stage to predict and plan the AV's route, behavior, and movement. This iterative decision-making process involves generating and selecting the optimal prediction while considering relevant information, including real-time map data. This process is influenced by various constraints.
- **Vehicle platform manipulation**: This phase focuses on the functional components of the control and chassis for AV movement and stability during autonomous operation. As a closed-loop vehicle navigation controller, the control module translates high-level plans generated by the decision and planning module into low-level actions, issuing commands to the vehicle's actuators (steering, braking, and acceleration) to follow the predetermined trajectory. Meanwhile, the chassis component bridges electrical and mechanical systems to ensure safe navigation, interacting with mechanical parts such as the accelerator pedal, brake pedal, steering wheel, and gear motors.

Nowadays, with substantial expected benefits and effects, research and development efforts are in progress to ensure the large-scale deployment of ADS. This involves refining the design and testing of ADS for a transition from prototype to production, focusing on understanding their broader implications. Recognizing and addressing these implications is crucial for unlocking the full potential of ADS, ensuring safe society integration, and maximizing societal benefits. Key implications of ADS are highlighted below [61]:

- **Safety advancements**: Ensuring the safety and security of ADS is a primary concern. The challenge lies in designing safe and secure systems while addressing potential drawbacks such as traffic congestion and functionality bottlenecks.

Therefore, careful architectural design, certification, testing, verification, and validation are pivotal in assessing the safety and security of ADS. These steps are essential for facilitating the smooth integration of AVs into the transportation ecosystem.

- **Technological innovation**: Given that ADS advancement relies on cutting-edge technologies, deployment pathways need to be flexible and adaptive. This flexibility accommodates potential changes in funding availability, future research findings, and broader societal needs and trends impacting the rate of technological adoption. These pathways also facilitate the testing and demonstration of new technologies, paving the way for the eventual commercial deployment of ADS-equipped vehicles.

- **Regulatory and legal framework**: Emerging trends and needs for high-level ADS, based on existing mature systems, emphasize the importance of working towards global harmonization of regulations and standards for autonomous driving. This involves revising and modernizing regulations and laws to eliminate unnecessary barriers to integrating advanced car designs and features proposed by autonomous driving systems. Additionally, developing safety-oriented frameworks and techniques is crucial for evaluating the performance of ADS technology.

- **Infrastructure upgrades**: Supporting AV deployment and ensuring their safe and secure operation require significant improvements to current road infrastructure. This includes enhancing road markings and implementing advanced communication systems. Additionally, the infrastructure must address potential impacts on the entire transportation ecosystem that may arise with the introduction of AVs.

- **Redefined user experience**: The advancement in ADS-equipped cars promises enhanced convenience for passengers but also introduces complexity in design due to human factors and behaviors. Human-machine interface (HMI) systems need human-centered designs to enable an effective understanding and interaction with users based on their unique characteristics and personalities. The public needs to adapt to and accept the shift towards autonomous driving, showing a willingness to invest in automation features and the concept of owning an automated vehicle.

- **Ethical considerations**: ADS introduction raises normative ethical issues related to the AV industry, environmental and public health ethics, and decisions made by ADS algorithms. Ethical debates and guidelines are needed, addressing how the system prioritizes safety in critical situations.

- **Public acceptance**: ADS technology needs to be widely accepted and trusted as a prerequisite for deployment to be successful. Public education and awareness campaigns must balance divergent opinions on the technology's practicality and acceptability in the marketplace.

While the previously discussed implications are important, the significant progress in automated driving technologies currently overshadows the existing challenges related to deployment and implementation, which may be difficult to address in the near term, as discussed in the next subsection.

1.2.4 Challenges of Autonomous Driving

Despite the substantial benefits of autonomous driving, several challenges and open issues require the research and industry community's attention for a mass production perspective. This section examines key challenges of autonomous driving from various perspectives [62, 63, 64, 65, 66, 67, 68, 69, 70].

- Technology-related issues:

- **Sensing and perception**: Safe navigation depends on the AV's capacity to understand its environment through vehicle perception, as outlined in earlier subsections. Given the dynamic and unpredictable nature of a vehicle's environment, as well as challenges in sensor technology, sensor fusion, and sensor data, developing effective and reliable perception algorithms requires accurate detection, interpretation, and semantic assignment to visual cues. This includes on-road object detection and tracking, contextual information awareness, drowsiness monitoring, anomaly detection, and traffic sign recognition.
- **AI and machine learning**: AVs rely on AI algorithms, like DL-based predictive models, to perform complex rational, intelligent, real-time driving tasks. However, these AI-powered applications face challenges, including limited driving scenarios datasets, sparse and costly annotations, training inefficiencies, and handling corner cases such as anomaly data, outliers, and out-of-distribution (OOD) data, with a lack of effective evaluation metrics. Training models with such datasets struggles with scalability, robustness, and adaptability. Promising approaches to address these issues include online learning strategies, reduced data dependency learning methods, human-in-the-loop AI integration, and improved mobility-related decision-making. Further exploration in areas like emotional intelligence and complex social interactions is needed.
- **Planning and control**: Path and motion planning for autonomous driving technology face several challenges, including simulation validation dependencies, data processing, real-time performance, and safety assurance. Despite DRL advances, issues such as balancing exploration and exploitation and bridging the simulation-reality gap remain. Future research aims to enhance planning strategies for incomplete perception data, improve planning quality and consistency,

interpret learning-based planners better, and explore cooperative multi-agent planning. Vehicle control algorithms need to address safety, vehicle interactions, and complex dynamics. Promising areas for development include cooperative control in the IoV, multi-objective optimization, coordinated control under uncertainty and delays, fault-tolerant techniques, and real-world traffic testing. Moreover, improving DL-based vehicle controllers in terms of computational load, architectural constraints, and goal specification is necessary, with the potential exploration of remote control and energy-aware frameworks.

- **Mapping and localization**: High-definition maps play a crucial role in the localization of autonomous cars, but they present several issues. Developing universal mapping formats fulfilling algorithm requirements, providing necessary static and dynamic information, determining the minimal data quality required for safe driving, and assessing the impact of poor data, is complex. Additional challenges include storage, updating, dissemination, and ensuring privacy and trust in mapping data within shared frameworks, particularly when integrating data from multiple open and commercial map sources. Developing a common mapping standard is essential for transitioning from testing to real-world autonomous driving.

- **Testing and validation**: Comprehensive testing and validation are crucial for ensuring the safety and reliability of AVs before mass production. Nevertheless, developing standardized test protocols and scenarios mimicking real-world environments and defining evaluation criteria for computer-simulated testing pose significant challenges. Bridging the gap between simulated and real test environments remains an unresolved issue.

Safety-related issues: Overcoming autonomous driving safety challenges related to hardware failures and software malfunctions is a primary concern for the research and industry community. The most likely issues in this regard are explored below [71, 65, 72, 73, 74, 75, 76, 77]:

- **Functional safety and technical safety**: The overall safety of critical autonomous driving systems depends on reliable hardware, software, and architectural components. Ensuring the functional safety of DL-based systems is particularly challenging due to potential malfunctions and failures at various vehicle operation levels, probabilistic error rates, and the inherent nature of deep learning models. Various approaches have been proposed to address these challenges, including risk assessment in real-world environments, safety analysis to identify underlying concerns and objectives, policy functions and control strategies to ensure safe operation, thorough model accuracy and uncertainty evaluations, and real-world testing. Additionally, addressing dataset completeness, fairness in learning, context awareness, and compliance with regulatory constraints remains an active area of research. Specifying technical safety requirements for system interfaces, environmental constraints, and failure detection mechanisms is also crucial to ensure overall safety according to ISO 26262 standards.

- **Privacy and security**: Given that AVs generate and collect significant data in public places, concerns exist about sharing knowledge while balancing sensitive passenger data security and privacy. Protecting vehicle occupant privacy and ensuring security involves addressing vulnerabilities in hardware and software systems, detecting intrusions, and securing real-time and historical on-board data collection, transmission, and storage. This is crucial to prevent attacks, and unauthorized access, and to maintain data confidentiality, integrity, and authentication. Since DL-based algorithms embedded in the AV software architecture are vulnerable to threats, ensuring robustness against adversarial attacks is essential.

- **Failure detection and diagnostics**: Safety-critical driving scenarios require robust fail-safe mechanisms to effectively handle operational failures or emergencies beyond functional and technical safety. Addressing failures and diagnosing issues poses significant challenges, particularly in developing fallback strategies for safe human control transitioning and establishing safe recovery protocols. Challenges include the lack of standardized definitions for sensor failures, insufficient research on sensor failure detection, difficulties in detecting sensor data failures even when the sensors are functioning properly, and algorithm failures in complex scenarios. Addressing these challenges involves developing advanced algorithms to improve performance and mitigate software design errors.

- **Collision avoidance**: The ultimate objective of autonomous driving collision avoidance is to develop technologies emphasizing safety first and preventing accidents through accurate collision prediction and correct avoidance decisions. Even with the progress made in AI and V2X communication, effectively handling various types of collisions remains an unanswered question. These challenges include addressing the limitations of deep learning techniques in terms of performance, mimicking human driver cognition, achieving high automation levels in critical situations, and shifting towards cooperative collision avoidance schemes.

- Broader challenges: In addition to the aforementioned challenges, further issues remain encompassing broader aspects of autonomous driving. Some of the key challenges include the following [22, 78, 63, 71, 79, 80]:

- **Fairness and transparency in DL for AD**: Recent concerns emphasize the necessity of explainability in DL-based models for perception, decision-making, and action or control. Due to the black-box nature of these models and outcomes uncertainty, explainable AI can enhance fairness and transparency in autonomous driving systems by interpreting what the model has learned, identifying sources of bias, and analyzing decision-making processes. However, achieving explainability presents several challenges, including issues related to model and algorithm selection, safety assurance, trust and user acceptance, legal and ethical responsibilities, human-machine interaction, regulatory compliance, training, and debugging.

- **Complexity and uncertainty**: AV embedded systems involve multiple decision-making components whose interactions can be challenging to understand, even for experts. These components, often relying on AI models, increase system complexity and exhibit uncertainty due to computational challenges, lack of explainability, and "black-box" nature. Ongoing debates propose various methods for reducing both epistemic and aleatoric uncertainty. Balancing complexity and uncertainty is crucial to developing an efficient autonomous driving system and requires further investigation.
- **Robustness and adaptability**: Since most AI algorithms utilized for autonomous driving are learning-based, their performance heavily relies on data reflecting specific environmental conditions. However, accuracy remains substantially low, as the dynamic nature of road scenarios compromises the effectiveness of various autonomous driving tasks. Consequently, systems designed for AD-related tasks must possess robust adaptability to adjust seamlessly to changing environments.
- **Big Data and real-time processing**: Ensuring the safety of AVs involves continuously acquiring substantial quantities of data of varying types and quality through an array of sensing technologies. This practice generates vast quantities of big data, essential for the vehicle's comprehensive environmental awareness. However, real-time processing of such extensive and diverse datasets presents significant challenges, including ensuring data accuracy, minimizing power consumption, and managing costs. To address these challenges, an intelligent data prioritization system is necessary to recognize and rank a wide range of data, retaining only the most relevant data for further analysis while discarding unnecessary data.
- **Connectivity and communication**: The emergence of AVs has created opportunities for more advanced systems beyond ego-only systems (or standalone vehicles) to include cooperative driving automation (CDA), connected vehicles, and connected autonomous vehicles (CAV). These developments highlight the critical role of connectivity and cooperative decision-making in improving performance. A significant technical challenge is ensuring interoperability among diverse autonomous systems from various manufacturers, which necessitates the development of unified standards. Next-generation wireless networks, particularly 5G and forthcoming 6G

technologies, are pivotal in providing the necessary environmental awareness for vehicles through ultra-reliable and low-latency communications. These technologies are vital for the broad deployment of CAVs, enabling smoother vehicular communication. However, integration of 5G, beyond 5G (B5G) technologies, and AV technologies is still in its early stages, with numerous research challenges to overcome, including software heterogeneity, validation, verification, and latency issues.

Conclusion

This chapter has provided a comprehensive overview of autonomous driving technologies. We presented several concepts related to the evolution of this field, such as ITS and VANET, highlighting road safety approaches and their impact. Subsequently, we explored the historical development and current state of AV technology, introducing various architectural designs and implications of AV-integrated systems for autonomous driving deployment. Finally, we concluded the chapter by discussing the challenges associated with these systems.

In the next chapter, we will explore concepts related to building object detectors, which are at the core of an AV's perception systems. Our focus will be on the foundations, development approaches, and challenges involved in deploying secure and safe AVs.

Chapter 2

Object Detection in Autonomous Driving

Introduction

As discussed in the previous chapter, the development of autonomous driving technology heavily relies on the vehicle's capability to accurately perceive and interpret its environment. Object detection represents a core component of vehicle perception systems and is essential for ensuring safe and reliable navigation. This technology allows AVs to recognize and track objects such as other vehicles, pedestrians, cyclists, and road signs, facilitating informed decision-making. Innovative neural architectures have permitted large steps forward in building robust object detection systems for AVs. However, this progress has been accompanied by the challenge of expensive labeling and training processes. Moreover, achieving functional safety necessitates identifying deficiencies in hardware and software components and developing effective solutions to address these issues.

This chapter presents a literature review of advances in the field of object detection and their application to autonomous driving. The first section 2.1 of this chapter introduces fundamental concepts of object detection, tracing its historical development through traditional and deep object detection periods and describing various neural network architectures dedicated to this task. It also examines approaches to reducing the costs associated with designing deep detector models. The next section 2.2 outlines key elements in applying object detection within the context of autonomous driving, including commonly detected objects, sensor modalities, datasets, and evaluation metrics. This section also reviews state-of-the-art methods and approaches in this context and summarizes the challenges associated with object detection in autonomous driving, concluding the chapter. This structured analysis will set the foundation for developing more robust and real-time object detection systems while ensuring functional safety for AVs in subsequent chapters.

2.1 Object Detection

The previous chapter indicates that safety is the primary focus in designing autonomous vehicles while performing autonomous navigation. This is accomplished by guaranteeing safe and collision-free road travel through efficient, adaptable, and robust object detection and path-planning algorithms.

This section covers key concepts of object detection, including its principles, main categories of object detection strategies, the architecture of recent and popular frameworks, and the latest methods for label-efficient and cost-effective learning.

2.1.1 Object Detection: Principles

Visual recognition is a fundamental area of computer vision and machine vision, aiming to imitate human visual capabilities by developing artificial visual systems that can understand images. Two primary algorithms have been developed to interpret and categorize scene content: image classification and visual object recognition. While traditional object detection approaches primarily relied on matching and geometric verification paradigms for recognizing specific, salient objects, contemporary approaches employ statistical models of object appearance or shape, acquired through exemplar-based learning, for both generic (coarse-grained) and fine-grained object recognition tasks. These models can identify the presence, position, and size of all instances of a given object class in images, often represented by bounding boxes or pixel-level masks. Object detection tasks may also involve recognizing multiple categories, which presents challenges as complex multi-class problems.

More specifically, the object detection algorithm employs a fundamental visual pattern discovery process, using visual representation to automatically determine the probable locations of target objects. This process involves extracting relevant visual primitives that provide useful information about the location, size, and shape of objects. The analysis of these visual patterns can be conducted using either a top-down or bottom-up approach, as detailed below [81, 82]:

- **Bottom-Up approaches**: in bottom-up approaches, Visual pattern discovery begins by identifying basic visual primitives and progresses through merging these initial elements to identify more complex visual patterns. These patterns are considered collections of parts with local appearance. Local visual feature detectors initially identify object parts in sparse sets of possible locations. Subsequently, the candidate locations, which initially served as visual primitives, are combined using geometric or statistical data to suggest and assess potential object placements in the image space. By leveraging information about the relative placements of parts, the object detector attempts to consolidate component detections into coherent object detections. This is often achieved through local part detectors designed specifically for particular classes of parts. In addition to discriminative or generative statistical models, these approaches can use various methods to combine parts and handle partial occlusions and unexpected variations in object pose. However, they have limitations, including the relatively lower reliability of small part detectors compared to larger whole-object detectors, and the complexity of spatial reasoning required to efficiently combine sets of unreliable part detections, which may include numerous misses and false alarms.

- **Top-down approaches**: In top-down object detection approaches, images are modeled based on high-level visual patterns, representing the entire object by its appearance. These methods typically use one or more rigid object-shaped templates that are class-specific, along with intensive multi-scale scanning throughout an image pyramid to identify object instances and their extents. In most top-down methods, detectors are discriminatively trained to recognize relatively "rigid" and unoccluded object classes.

Regardless of the approach employed, the main advancements in object detection can be divided into two distinct historical eras: the traditional object detection period (before 2014) and the deep learning detection period (after 2014). During this transition, research findings indicated that the establishment of standard benchmark datasets could facilitate the building of accurate models for detecting various object classes, as detailed in the following subsections.

2.1.2 Traditional Object Detection

The adoption of machine learning algorithms has revolutionized visual object recognition, shifting the landscape towards statistical learning-based approaches. in this context, conventional generic category object detection approaches identify object categories similarly to image classification tasks. This is achieved in a supervised setting by collecting training examples for a target object class, extracting visual features from annotated samples, and using those features to train a detection model capable of predicting object localization in unseen images. Figure 2.1 illustrates the fundamental framework of traditional approaches.

Conventional object detection algorithms typically comprise three stages, as illustrated in Figure 2.1: object proposal generation, feature extraction and selection, and classification. During the online recognition stage, object proposals are initially generated from the input image as regions of interest, using an underlying object position hypothesis generator. Subsequently, visual descriptors capture visual representations of the candidate regions by extracting hand-crafted features. Through these extracted features, the trained classifier's decision values identify whether or not objects are

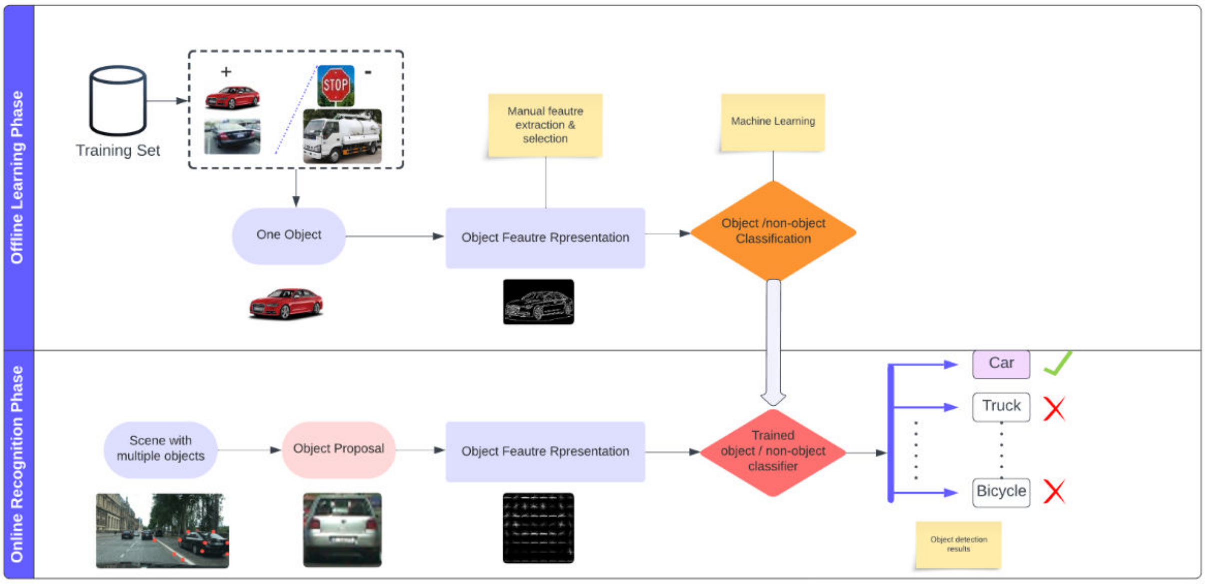

Offline Learning Phase
Training Set
+
-
STOP
Manual feautre extraction & selection
Machine Learning
One Object
Object Feautre Rpresentation
Object /non-object Classification
Online Recognition Phase
Scene with multiple objects
Object Proposal
Object Feautre Rpresentation
Trained object / non-object classifier
Car
Truck
Bicycle
Object detection results

Figure 2.1: Pipeline of traditional generic object detection algorithms.

present in the unseen input image. Leveraging machine learning and supervised learning techniques, this object/non-object classifier is built by learning decision rules from the extracted features of a representative training set.

Formulated as a classification problem, these traditional algorithms can perform real-time detection but at the cost of limited accuracy and heavy computational load. Their performance and effectiveness can be significantly influenced by the choice of training data, the underlying feature representation, and the selection between discriminative and generative classifiers.

A key factor for improving performance is the extraction of visual representation. Human-engineered visual feature descriptors, such as Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG), and Scale Invariant Feature Transform (SIFT) are commonly used to encode object appearance based on color, texture, and shape properties. With the possibility of incorporating segmentation or contextual cues, local and global features extracted from these descriptors play a crucial role in traditional object detection, helping to distinguish object instances from non-object instances [4, 83].

Another factor is the choice of statistical models for object appearance that can handle variations in appearance and shape within the same category. Discriminative models are generally preferred due to their ability to directly model the posterior distribution of class categories given input features, leading to more accurate object detectors compared to generative models [4].

As reviewed in the literature [81, 4], current traditional approaches are typically classified into part-based and window-based techniques. The relationship between these techniques can be viewed as a balance between representational complexity and search effort, as explored below:

- **Part-based approaches**: part-based object models, which belong to bottom-up approaches, use a combination of fixed parts and flexible spatial arrangements to represent objects. These models were initially built using pre-defined templates based on whole objects, composed of manually created part templates using pixel-level features. A spring-like mechanism was employed to reduce small shape distortions to fit these parts together, taking into account differences in the parts themselves and their spatial positions. In contrast, more recent techniques aim to learn the appearance of parts directly from training data. The core methodology still involves finding object segments or parts with similar appearance and spatial configurations. These parts and their spatial relationships are then used to generate object models. More efficient part-based object recognition models have emerged from the development of local invariant features. This approach enables robust object discrimination across scales and complexities by exploiting a voting system across different object parts. However, dealing with the high combinatorial complexity of these models remains a challenge. Various spatial modeling techniques have been explored to address this issue, each offering different levels of computational requirements and connectivity. Among the most notable are the Bag of Visual Words, Constellation, Star, Tree, k-fan, Hierarchical, and Sparse Flexible Models. Partial occlusions are easier to handle with these approaches, but their practical effectiveness depends critically on several factors related to the training process of the learned model, including the type of supervision (full or weak) for the learning algorithm, the number of training examples and the extent of labeling within these sets. Other knowledge, including object location, scale, and any uncertainty, can be derived or learned directly from the annotations provided, such as hand-labeled part locations.

- **Window-based approaches**: Window-based approaches, as part of top-down strategies, process images by considering the whole object's appearance. The basic sliding window-based detector pipeline is summarized in Figure 2.2. Initially, the input image is resampled into a pyramid structure using a sliding window technique that systematically scans the image at various positions and scales, forming a pyramid of multi-scale sub-windows considered as candidate object proposals. Robust visual features are then extracted from each sub-window using human-engineered feature descriptors, creating a feature pyramid. This structure is explored, and features are evaluated by a classifier at multiple scales to determine the presence of an object, leading to the detection of objects of different sizes by generating bounding boxes for potential objects. To obtain the final detections, the classifier's decisions are refined through a post-processing step such as non-maximal suppression, which consolidates overlapping boxes and eliminates redundancies based on a threshold. Various discriminative classifiers are used in this context, ranging from simple decision trees to more complex models like convolutional neural networks (CNNs) and Support Vector Machines (SVMs), each chosen based on the specific requirements of the detection task. Sliding window-based detectors currently show superior performance for deformable object classes and classes with significant intra-class or viewpoint variability. However, they have several disadvantages, including the computational cost of the window-level method, the loss of context from the overall scene, and the use of rectangular, axis-aligned windows that may not be suitable for many object categories with non-box-shaped appearances, even for partially occluded objects. Additionally, data-related bottlenecks in training and testing, the quality of the classifier, and the granularity of sampling steps during scale and position sweeps have also impacted the effectiveness of sliding window detectors.

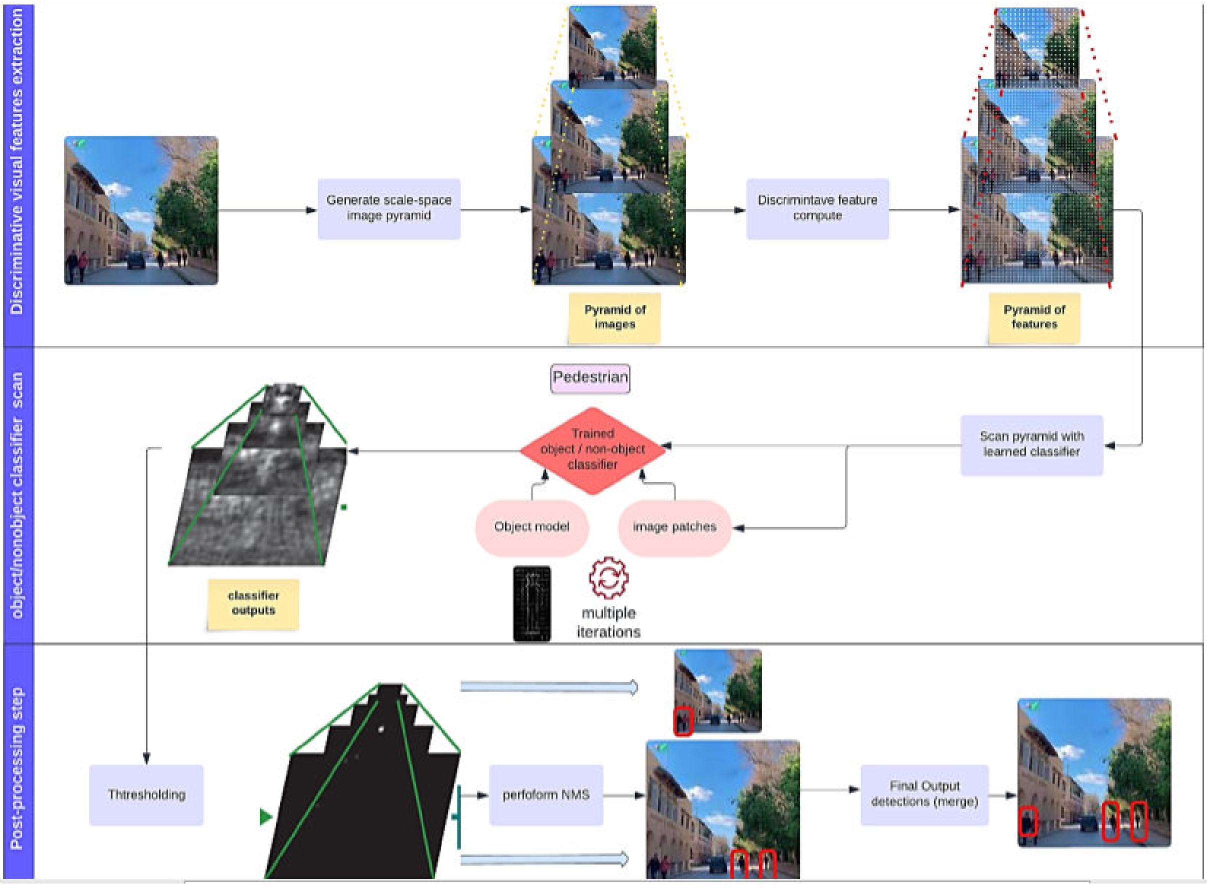

Discriminative visual features extraction
Generate scale-space image pyramid
Pyramid of images
Discrimintave feature compute
Pyramid of features
object/nonobject classifier scan
Pedestrian
Trained object / non-object classifier
Scan pyramid with learned classifier
Object model
image patches
multiple iterations
classifier outputs
Post-processing step
Thtresholding
perfoform NMS
Final Output detections (merge)

Figure 2.2: An overview of gradual refinement in localization from coarse to fine by the sliding window detection approach.

However, the limitations of traditional object detection methods have motivated the exploration of more advanced techniques. In recent years, the field has experienced a significant evolution, particularly driven by the emergence of deep learning, which has revolutionized numerous areas of computer vision. Consequently, research has increasingly focused on deep learning-based approaches for object detection, promising improved accuracy and efficiency.

2.1.3 Deep Object Detection

Due to the abundance of innovative computer vision applications driven by deep learning, research on object detection has recently shifted toward deep learning-based approaches. Deep object detection is a multi-task problem that integrates both object category classification and boundary box regression to perform object recognition and localization. This is typically achieved using standard deep neural network architectures, a form of non-symbolic artificial intelligence, which are used either as feature extraction or inference algorithms as illustrated in Figure 2.3. Figure 2.4 depicts the general pipeline of deep object detection training.

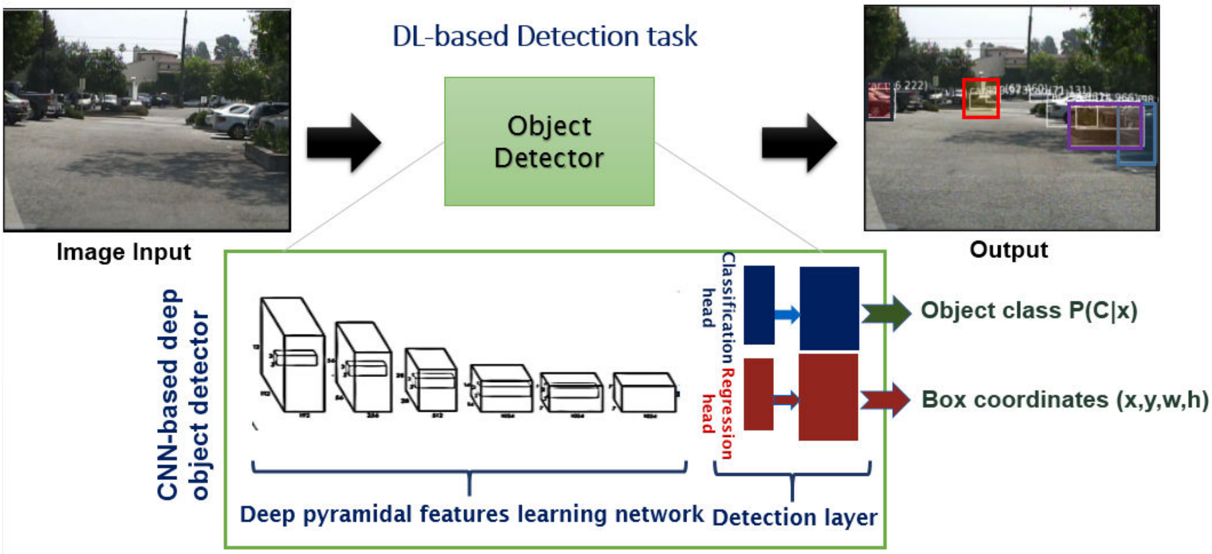


Figure 2.3: An overview of the deep learning-based object detection task.

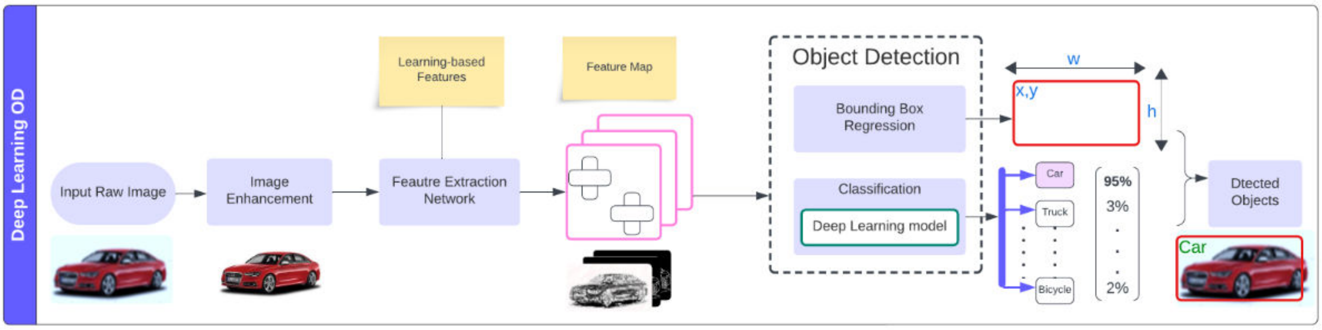


Figure 2.4: An overview of the end-to-end deep neural object detection pipeline.

As shown in Figures 2.3 and 2.4, the deep object detector is trained in an end-to-end scheme. In contrast to traditional approaches, the human-engineered feature extraction stages are progressively being replaced by learning-based feature extraction networks, also known as backbone models, such as CNNs. These models aim to automatically learn feature representations from raw image pixels, enabling the extraction of high-level object features and the generation of feature maps for accurate object identification. The classification task can be performed using the same backbone, a separate DNN, or a conventional machine learning technique, leading to the construction of a robust and real-time deep object detector, although at the cost of lower accuracy compared to traditional object detectors [83, 84, 85].

In practice, the most popular methodologies used to train a deep object detection model are as follows:

- **Training a model from scratch**: This method relies on two key factors: (1) a large amount of labeled training data, and (2) the design of a network architecture, including layers, weights, and other parameters.
- **Using a pre-trained deep learning model**: This approach involves a transfer learning paradigm, where a pre-trained model like AlexNet or GoogLeNet is fine-tuned as a baseline with a new dataset that includes previously unseen classes. The training process using this method can converge more quickly and produce more efficient results by leveraging the knowledge acquired during the baseline model's initial training on a large dataset.

In addition to the training of deep models, the design of their architecture is crucial for the successful application of deep learning, impacting both the development and deployment of real-time applications. Achieving high target task accuracy and low latency inference on the target platform is essential for these applications. To automate and optimize the design of deep model architectures, Neural Architecture Search (NAS) has been used to develop low-latency networks for various vision tasks [86]. Within the field of automated machine learning (AutoML), NAS is closely associated with hyperparameter optimization and meta-learning, aiming to build networks that outperform the best hand-designed architectures. This approach minimizes the time model developers have to dedicate to data preparation, algorithm selection, and fine-tuning [87].

More specifically, NAS focuses on determining the "best fit" architectures for artificial neural networks (ANNs), which are widely used in machine learning. It employs various methodologies, including gradient-based techniques, evolutionary algorithms, and reinforcement learning. NAS approaches are classified according to the search space, the search strategy used, and the performance evaluation strategy, as outlined below [88]:

- **Search space**: This outlines the complete range of possible architectures and the type(s) of ANN expected to be developed and optimized.
- **Search strategy**: This describes the optimization method employed to navigate through the search space to identify an efficient architecture.

- **Performance estimation strategy**: This refers to any technique used to quickly assess the potential performance of a proposed ANN based on its design, without the need to build and train it.

Recently, advanced NAS technology has been well-tailored for deep learning to optimize deeper neural networks, especially in object detection. Aside from the utility of deep learning models for automated feature learning, automating the tedious step of designing the detection model architecture by exploring NAS is crucial. This involves encoding the entire architecture at one level and determining macro-level hyperparameters by exploring the macro-search space, rather than fully exploring cell-based or hierarchical search spaces. Additionally, training detection models from scratch is time-consuming and requires substantial GPU memory due to the complexity of the architecture [88].

In the literature, the use of NAS to optimize DNN architectures for object detection is rarely explored. In [89], the authors constructed research spaces based on state-of-the-art image classifier networks to develop improved backbones for object detection, taking advantage of both pre-training and fine-tuning for detection. Once the pre-trained backbone architecture is fine-tuned on detection datasets, the architecture search is carried out on the trained supernet. This process incorporates selected detection-specific elements from successful dense prediction architectures and utilizes an evolutionary algorithm for optimization. Conversely, the author in [90] proposed fast versions of NAS suitable for one-stage object detection macro-architectures. Their method aimed to search both feature pyramid networks and prediction head architectures while caching the features generated by a backbone to speed up architecture search.

With the maturity of deep neural networks for object detection tasks, numerous deep object detection models have been developed over the years. These models typically share the common architecture shown in Figure 2.5. A deep object detection model consists of a backbone network and a head network. The backbone network processes the raw input image to extract features, which are then fused and enhanced through the neck. The head network uses these extracted features to predict classes (labels) and the coordinates of bounding boxes surrounding the identified objects. Additionally, the design of network architecture for these models can be approached from two different perspectives: micro-architecture, referring to a "small" network unit, and macro-architecture (meta-architecture), referring to the entire network composed of these small units.

To provide a comprehensive overview of the current state-of-the-art, Figure 2.6 presents a detailed taxonomy of mainstream deep object detection algorithms, focusing on their architectural exploration [83, 84, 85].

As shown in Figure 2.6, deep object detection algorithms can be categorized based on various criteria. The following subsections outline these categories.

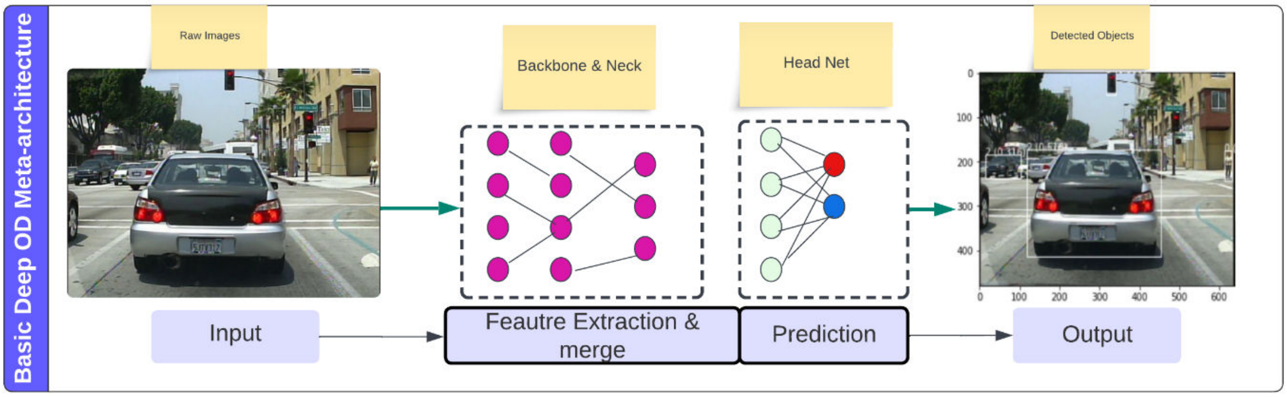


Figure 2.5: Basic process of deep object detection algorithms.

2.1.3.1 Framework-Related Classification

The primary distinction between two-stage and single-stage detection frameworks, as classified according to the framework criterion in Figure 2.6, is based on the composition of the head network in each framework's meta-architecture and the impact of the object detection pipeline on the balance between accuracy and detection (inference) speed metrics. Figure 2.7 clearly illustrates the basic meta-architecture for each framework, highlighting the differences between single-stage and two-stage detection algorithms as explained below.

- Two-Stage (two-shot) framework: In two-stage object detection models, the head network is a multi-stage model consisting of two separate networks: the region proposal module and object detection module, as shown in Figure 2.7(b). The first module uses the feature maps generated by the backbone network, which processes the raw input image, to identify several regions of interest (RoI). Subsequently, these candidate regions of objects are processed by the pipeline for object classification and bounding-box regression. Consequently, the objects within the region proposals can be classified, and their localization can be refined. Two-stage frameworks primarily leverage selection techniques, such as Edge Boxes or Selective Search, in conjunction with the NMS post-processing for final detection. Selection algorithms generate region proposals, while NMS rapidly eliminates redundant prediction boxes [91, 92, 93]. Popular two-stage frameworks include the R-CNN series, R-FCN, and SPPNet.

Candidate-based two-stage algorithms can achieve relatively high accuracy and more reliable detections. However, this category suffers from slower inference time and detection speed. These shortcomings originate from several factors, including the required fixed size of the input image, the time-consuming nature of the region proposal generation and post-processing steps, the computational complexity of multi-stage processing, and the scale of the large models and feature space extracted. Consequently, these limitations can make two-stage algorithms inappropriate for real-time applications

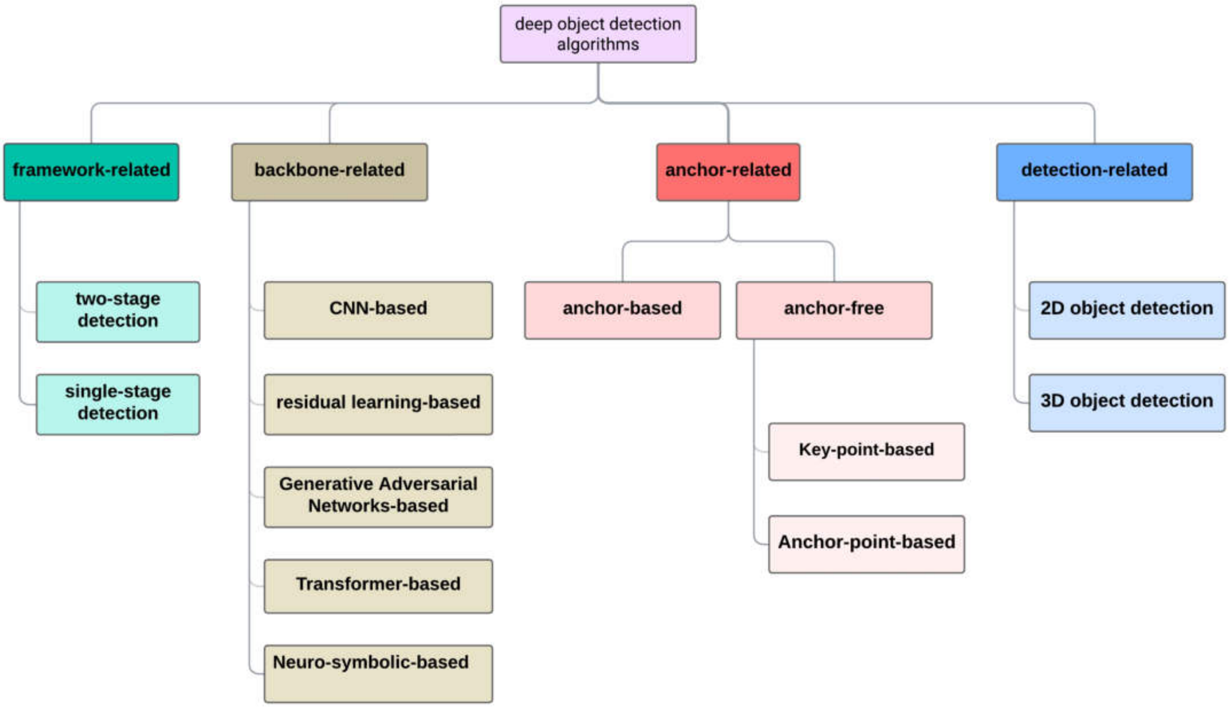


Figure 2.6: Taxonomy of deep object detection algorithms.

[94, 92].

- One-Stage (single-shot) frameworks: In contrast to two-stage frameworks, one-stage object detection frameworks (or regression detection model), as shown in Figure 2.7(a), skip the region proposal stage and instead fully integrate the region proposal module with the classification and localization computation, forming an end-to-end object detection model. In a single-shot detection, a single feed-forward neural network model delivers object classification probability and bounding box coordinates regression together at the same stage [94, 92, 95]. SSD, CenterNet, RetinaNet, and YOLO series are representative one-stage object detection algorithms.

By performing all computations through a single stage and optimizing the entire pipeline in an end-to-end manner, one-stage detection pipelines can predict results directly with a notably rapid detection speed. This is achieved by avoiding the explicit production of region proposals beforehand and employing a less computationally intensive regression analysis technique. However, they are less effective when dealing with overlapping and occluded objects and when handling strong foreground-background contrast and imbalances in positive and negative samples. These issues can prevent one-stage algorithms from achieving the same level of accuracy as two-stage algorithms [94, 91, 95].

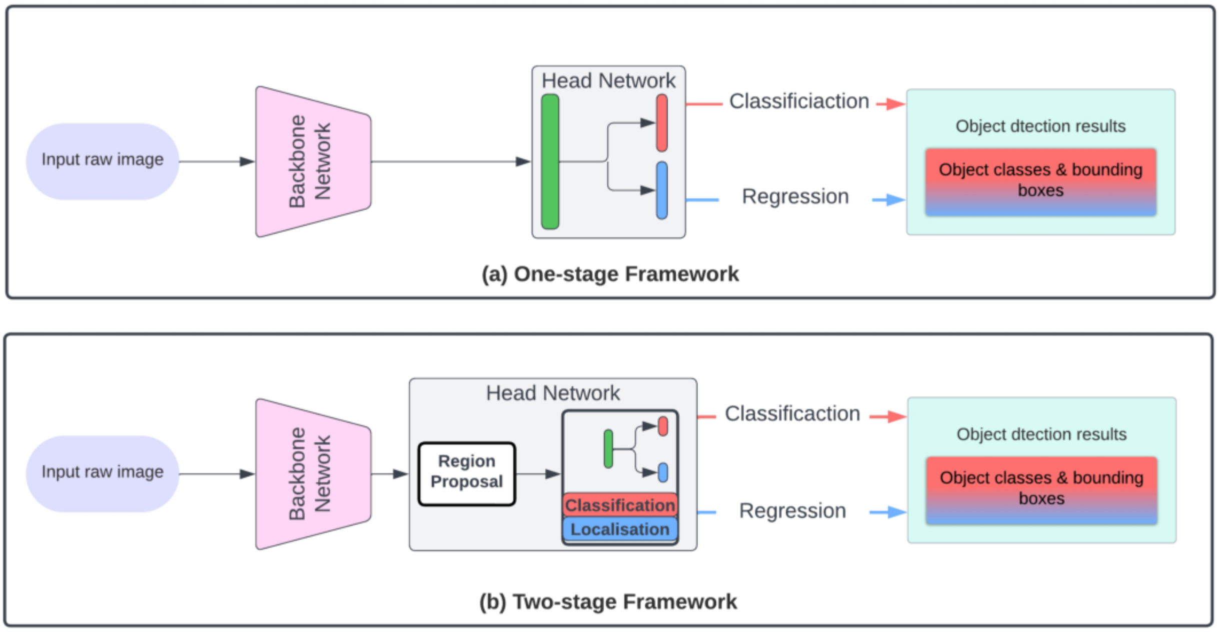


Figure 2.7: Basic structure of deep object detection frameworks.

2.1.3.2 Backbone-Related Classification

This subsection provides a detailed classification of deep object detection algorithms based on the integrated "small" network unit used as a building block of the backbone of the detection pipeline. These units include CNNs, transformers, residual networks, Generative Adversarial Networks(GAN), and neuro-symbolic models, as explained below.

- **CNN-based methods**: With the availability of significant computing resources and plenty of large-scale datasets, CNN-based deep learning algorithms have become the state-of-the-art approaches for object detection. As a non-symbolic artificial intelligence technique, CNN (or ConvNets) are deeper feed-forward neural networks frequently used in backbone networks to learn hierarchical feature representation at different levels, facilitating pattern recognition. Two varieties of CNNs are distinguished according to the image input dimension: two-dimensional (2D)-CNN and three-dimensional (3D)-CNN. The typical architecture of CNNs, depicted in Figure 2.8, consists of iterations of a sequence of layers, including (1) the input layer, (2) the convolutional layer, (3) the pooling layer, and (4) the fully connected layer.

This architectural structure is outlined in greater detail below [2, 96, 97]:

- **Convolutional layer**: As a core layer in CNNs, the convolutional layer integrates both linear and nonlinear operations through convolution and activation functions.

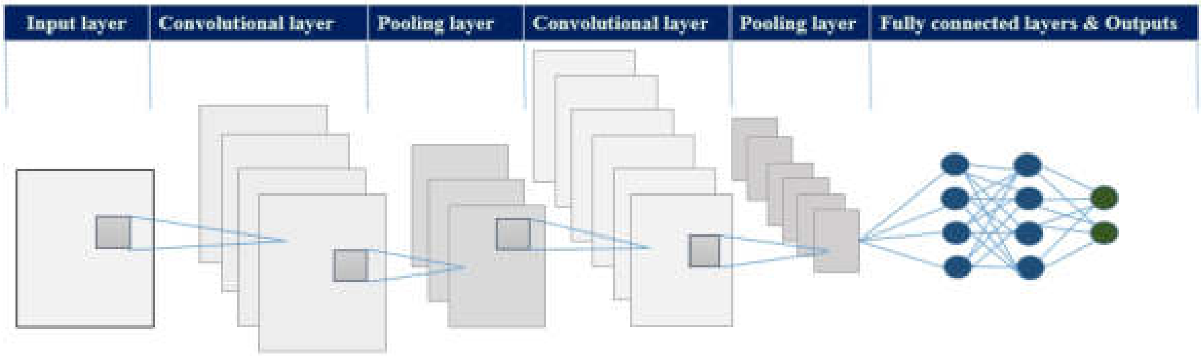
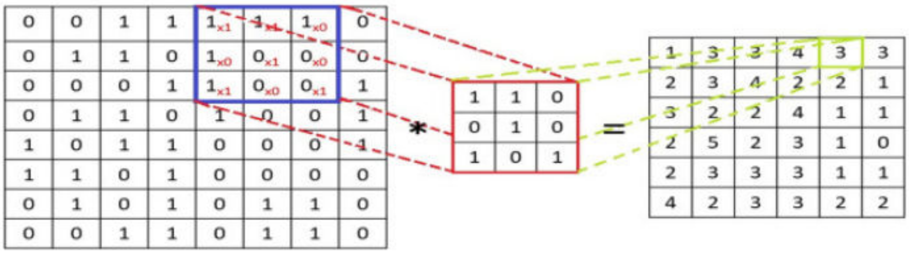


Figure 2.8: Basic CNN architecture.

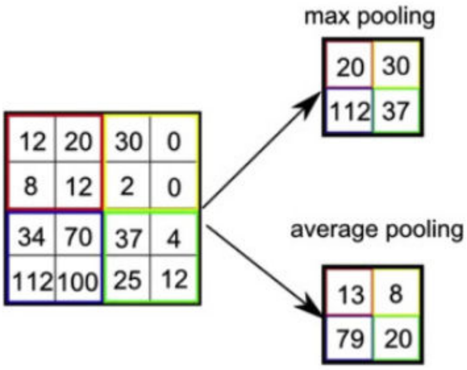
Convolution involves applying learnable filters (convolution kernels) to input tensors, with hyperparameters such as filter size and number determining the depth of the output feature map (2D activation map). Additional parameters like padding and stride control the alignment and spacing of the kernels. Each neuron within a convolutional layer is engineered to convolve a specific kernel with a limited subset of the outputs from the previous layer, perceived within its "receptive field." The process of computing the activation map is depicted in Figure 2.9 and can be described by this formula: Activation map $=$ Input $*$ Filter $= \sum_{i=0}^{\text{columns}} \sum_{j=0}^{\text{rows}} \text{Input}_{i-\Delta_i, j-\Delta_j} \times \text{Filter}_{\Delta_i,\Delta_j}$ [98]. Common convolution types include ordinary, transposed, hole, and depth separable convolutions, all featuring weight sharing across image positions for pattern learning. Activation functions like ReLU, RReLU, ELU, sigmoid, and tanh add nonlinearity to CNNs, aiding in learning complex data patterns and nonlinear models.



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1_{\times 1} & 1_{\times 1} & 1_{\times 0} & 0 \\ 0 & 1 & 1 & 0 & 1_{\times 0} & 0_{\times 1} & 0_{\times 0} & 0 \\ 0 & 0 & 0 & 1 & 1_{\times 1} & 0_{\times 0} & 0_{\times 1} & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 3 & 4 & 3 & 3 \\ 2 & 3 & 4 & 2 & 2 & 1 \\ 3 & 2 & 2 & 4 & 1 & 1 \\ 2 & 5 & 2 & 3 & 1 & 0 \\ 2 & 3 & 3 & 3 & 1 & 1 \\ 4 & 2 & 3 & 3 & 2 & 2 \end{bmatrix}$$

Figure 2.9: Convolutional layer [98].

- **Pooling layer**: Typically positioned between successive convolutional layers, the pooling layer involves downsampling (or subsampling) operations aimed at reducing the dimensionality of feature maps by aggregating groups of outputs from the previous layer into a single neuron. This process eliminates redundant convolutions while preserving crucial information. This reduction in dimensionality helps decrease the number of learnable parameters in subsequent layers, thus mitigating overfitting and accelerating computation speed. Additionally, pooling introduces translation invariance to small shifts and distortions. Hyperparameters in pooling operations, such as filter size, stride, and padding, are similar to those in convolution operations. Notably, pooling layers do not contain any learnable parameters. Typical pooling layer operations encompass max-pooling, average pooling, and Spatial Pyramid Pooling. For example, an average pooling layer computes the average of its input values, while max pooling selects the maximum value (see Figure 2.10).



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

average pooling

13	8
79	20

Figure 2.10: Types of pooling layers[99].

- **Fully connected layer**: The fully connected (FC) layer, positioned after the convolutional and pooling layer blocks, maps the extracted features to the network's final outputs, functioning similarly to a traditional neural network. Typically, the output feature maps from the last block of convolutional and pooling layers are flattened into a one-dimensional array and then fed into one or more fully connected layers, also known as dense layers. Each dense layer uses trainable weights to establish connections between inputs and outputs, followed by a non-linear activation function like ReLU. Depending on the learning task, the final fully connected layer acts as a predictor, with the number of nodes adjusted to match the desired target outputs and the activation function selected accordingly. Common activation functions for the final layer encompass sigmoid for binary and multiclass classification, softmax for multiclass single-class classification, and identity for regression tasks involving continuous values. For example, in multiclass classification, the last layer acts as a classifier with output nodes matching the number of classes, and the activation function, typically sigmoid, predicts probabilities for each class.

Training ConvNets involves the iterative adjustment of learning parameters, such as the weights of convolutional and FC layers, using backpropagation and gradient descent optimization. In each iteration, the model's performance is assessed by optimizing the loss function, or cost function, by finding parameter values that minimize the discrepancies between predicted outputs and ground-truth labels in the training dataset, thereby achieving low error rates. Adam and stochastic gradient descent (SGD) are well-known optimization techniques in the optimization and gradient descent classes [2, 96, 100].

CNN-based backbone networks have several characteristics. The stack of layers aims to capture feature representations that transform low-level pixel data from raw images into a discriminant high-dimensional data space, comprising high-level multi-resolution feature maps that integrate both higher-level details like objects and lower-level data like edges. In the same layer, the number of convolution kernels and feature maps is equal. The extent of the receptive field of a feature map represents an area of the input image that has been transformed into pixels of the high-level feature map [95]. Additionally, these networks offer numerous advantages and are naturally superior for object detection tasks due to their multi-resolution feature maps, hierarchical downsampling design, and resilient parameter learning. Furthermore, resilient local feature extraction and outstanding performance via convolutional kernel matching are made possible by their inductive bias qualities, such as translation invariance, weight sharing, and sparse connectivity [85, 101].

However, optimizing these models poses challenges, as training may encounter issues such as getting stuck in local minima or converging very slowly. Additionally, typical problems like the vanishing gradient effect can be observed in these architectures, where gradients diminish significantly during back-propagation, leading to very small weight updates. Moreover, while CNNs excel at extracting features through sliding independent convolutional windows, they may struggle to effectively capture global feature information [85]. AlexNet and VGGs are commonly used CNN-based backbone networks for deep object detection meta-architecture like SSD.

- Deep residual learning-based methods: The deep residual network (ResNet) [102], a variant of CNN, introduces residual modules (or residual blocks) as a replacement for the traditional sequential stacking of convolutional layers. Each residual module allows the output from an earlier layer to be fast-forwarded and added to the output of a later layer within the block. This addition occurs before the application of a non-linear activation function, with this bypass connection referred to as a shortcut or skip connection. Skip connections facilitate easier learning for the network, enhancing its performance and mitigating the vanishing gradient problem. ResNet employs two types of residual modules: basic and bottleneck, with the building block illustrated in Figure 2.11 [96, 103].

For multiple reasons, the ResNet architecture, which has many stacked residual modules, offers significant advantages over conventional CNNs. Its deep structure allows it to produce various tiers of spatial representation and provides wide receptive fields that capture fine-grained pixel information. ResNet efficiently divides the responsibilities of classification and localization, improves computational performance at higher levels,

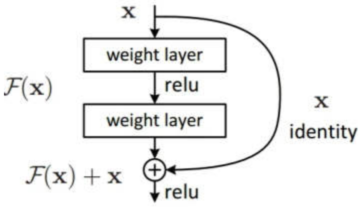

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.11: A building block of residual learning.

and utilizes effective encoding techniques with basic arithmetic operations. Despite its strengths, the current ResNet architecture has various shortcomings that need further investigation and improvement. These challenges include preprocessing issues like batch normalization and data augmentation techniques, architectural concerns such as the balance between computational efficiency and practicality, the implications of adding skip connections, and the trade-off between network depth and complexity. Additionally, there are still issues with training, such as the susceptibility to adversarial examples, the influence of biasing nonlinearities, and the potential impact of small local minima on training stability [96, 103].

- Generative Adversarial Networks-based methods: Generative Adversarial Networks (GANs) [104] are a type of unsupervised learning model that uses the principle of maximum likelihood along with adversarial training techniques for tasks like object detection [92]. It was initially employed for the generation of synthetic images. Unlike most recent deep learning methods, which often require numerous labeled instances to generalize well, GANs can be trained on data that is underrepresented or not included in the datasets, allowing it to learn about the underlying true data distributions [105].

In a GAN framework, a pair of separate ConvNets collaborate closely. The first is an unsupervised model, referred to as the generator "G", which is expected to generate adversarial examples that can trick the system by drawing Gaussian-distributed random samples. The second model, the discriminator "D", is a supervised model trained on both synthetic data produced by the generator and real data from a dataset. Its primary goal is to distinguish between real and generated data, effectively identifying and separating the artificial images from the real ones [106].

The GAN model's discriminator (D) and generator (G) undergo simultaneous training until they achieve a state of equilibrium where D is unable to discriminate between data produced by G and actual data. By employing an adversarial training strategy, this technique seeks to improve the detection network by generating occluded and distorted image samples using an adversarial network. An error signal is produced by the discriminator when it misclassifies a generated image. In response to this signal, the generator adjusts its parameters to improve the image. To optimize both models' parameters, adversarial learning requires them to be trained jointly. This is frequently done using optimization algorithms like Adam. These algorithms adjust the parameters of G and D to minimize their loss functions through the adversarial objective, similar to a two-player, zero-sum min-max game. Binary Cross Entropy (BCE) is commonly utilized as the loss function in the discriminator within GANs, serving to evaluate the discriminator's certainty in its classification of an image as either generated or genuine[105].

Compared to CNNs, which are more scalable, GANs are more suitable for achieving resilience in object scale variations. They are also quickly developing and finding use in a variety of GAN variants and frameworks reported in the literature. These variants and frameworks aim to address challenges in object detection tasks, including self-supervised object detection, learning from synthetic images, domain adaptation, detecting small-sized objects, feature generation, addressing object-level imbalance, and implementing data augmentation techniques [106].

- Transformer-based methods: The Transformer [107], a model that has recently gained prominence, exhibits increased flexibility to deal with massive datasets through its adaptive parameter learning process, unlike CNNs' static parameter learning. In the past decade, integrating transformers as the neck or backbone in various deep object detection frameworks has become a prominent research direction. The revolutionary accomplishments of transformer models in computer vision have significantly impacted this development [85].

One reason why transformer-based detection algorithms work so well at recognizing objects is that they leverage self-attention techniques to overcome some of the drawbacks of CNN-based deep object recognition models. Relative positional encoding is used to preserve translation invariance, and enough sets of heads are centered on every pixel in the convolutional receptive field for the self-attention process to resemble convolutional layers. This comprehensive attention operation effectively integrates local and global attention, seamlessly deriving attention weights based on feature correlations [**?**, 95, 101]. Another reason for the success of transformers is their capacity to comprehend and encompass intricate long-range relationships among objects, effortlessly acquiring comprehensive global information. Moreover, their scalability allows for training large models on extensive datasets without encountering performance limitations [108, 85].

However, in real-world scenarios such as autonomous driving, transformers have several drawbacks. Their substantial computational burden and complexity, influenced by factors such as storage requirement, energy consumption, performance trade-offs, and the ineffective use of transformers to represent image data in sequences, pose significant challenges. This necessitates the development of lightweight transformers for efficient deployment, along with the utilization of contextual object information and in-depth basic understanding to enable advancements in model optimization [101].

According to the literature, there are a couple of algorithms for object detection based on transformers: DETR-series and ViT-series [85, 95].

- Neuro-symbolic models-based methods: As a part of hybrid reasoning methods, neuro-symbolic Computing (NSC) aims to combine knowledge-based reasoning with ML. A subset of NSC is specifically designed to integrate DNNs with neuro-symbolic reasoning. These models support both "data-driven" deep learning, based on learning from complex data like images and sensorimotor data, and "knowledge-driven" symbolic reasoning, which uses complex representations of knowledge such as knowledge bases, semantic networks, and ontologies. This integration is achieved through a tightly coupled framework that facilitates enhanced object recognition capabilities[109, 110].

Current DL models typically learn object representations from high-dimensional raw features, which are often sub-symbolic distributed representations. In contrast, neuro-symbolic models aim to derive symbols from pixel data, utilizing symbolic representations instead of raw pixel space. Symbolic representation offers a compressed form of raw features to depict object attributes such as location, scale, and appearance in a more compact symbolic space. This approach enables high-level abstraction, reasoning, and learning, maintaining concepts and knowledge in a hierarchical and structured format. Symbolic knowledge and its associated reasoning can support deep learning model building in several ways. This integration can take place in the pre-processing stage for data augmentation performing, in hidden layers as components of the architectural layout or optimization functions, and in post-processing stages for model prediction verification [111, 109].

Neuro-symbolic models show potential in various areas: they can incorporate symbolic knowledge into neural representations or parameters, improving model controllability by aligning neural model behavior with high-level goals and symbolic representations. They also offer the possibility of more effective data utilization and can greatly enhance interpretability, generalization capabilities, robustness, and explainability [110]. Nonetheless, the symbol grounding problem, which deals with integrating highly abstract symbolic knowledge with neural representations obtained from actual raw input, such as text or images, remains a significant barrier to neuro-symbolic AI[112].

In the literature, a notable trend in neuro-symbolic approaches involves the integration of logic and neural models, known as logic-based neural models. The most typical neuro-symbolic, end-to-end object detection architectures are Logic tensor networks (LTN) [112] and its version Faster-LTN [113].

2.1.3.3 Anchor-related Classification

Generally speaking, there are two types of deep object detectors distinguished based on the assumption regarding prior anchor boxes, as detailed below [85, 114, 115]:

- Anchor-based methods: Anchor-based techniques employ an anchor mechanism designed to enhance the precision of bounding boxes and decrease computational overhead. This involves generating multiple candidate anchor boxes to characterize object dimensions, positions, and shapes in the image. These anchor boxes are then used to predict the category for each object and refine their coordinates to ultimately yield the refined anchor, which serves as the final prediction [114].

Anchor boxes are pre-defined rectangular shapes strategically placed across the feature map with specific dimensions and aspect ratios. These dimensions and ratios are specified to represent the range of object sizes and shapes present in the training set. During the detection phase, these anchor boxes are uniformly distributed over the image, using the same configuration for every image analyzed. The network predicts properties for each anchor box tile, including probability scores, background likelihoods, intersection over union (IoU), and positional adjustments, rather than the bounding boxes themselves. To ensure the model focuses on real objects, a threshold is specified for the estimated likelihood and computed IoU. Guided by the ground truth information, a loss function for every anchor box is calculated using these thresholds [92, 115].

Anchor-based detectors distribute anchors of varying sizes across different levels of a feature pyramid, enabling the detection of multiple objects, including those of different sizes or those that overlap [94]. Anchors act as references for regression and as candidates for classification, facilitating the generation of proposals in two-stage detectors and the determination of final bounding boxes in one-shot models[115].

Anchor-based methodologies represent a sophisticated technological approach, offering mechanisms for identifying positive samples crucial for classification and regression tasks. Nonetheless, configuring anchor boxes manually introduces a series of complications [92, 85]:

- **Detection across scales**: The reliance on hyperparameters, such as the dimensions and aspect ratios of anchor boxes, can adversely affect detection capabilities across different scales.
- **Model Generalization**: The reliance on artificially predetermined anchor boxes may compromise the model's ability to generalize across various scenarios.
- **Imbalance between classes**: A large portion of anchor boxes generated are negative samples, creating a significant imbalance between positive and negative samples.
- **Positioning and background noise**: Positive sample square anchor boxes may encounter issues with precise positioning and background feature interference.
- **Matching ground truth and IoU**: The methodology's effectiveness is hindered

by the scarcity of ground truth bounding boxes and the IoU matching process between these and the anchor boxes.

- Anchor-free methods: Anchor-free methods operate similarly to anchor-based techniques but without using carefully designed, pre-defined anchor boxes. Instead, they identify objects directly by focusing on a key, central point, or area (grid of cells) within the object [114]. Beyond the placement of multiple anchors at each location, the core concept of these strategies is to make dense predictions at the pixel scale, either by setting up initial anchor points or by leveraging key feature points to infer the bounding box coordinates. This reveals that anchor-free algorithms are generally classified into two groups: one focuses on key points and the other on anchor points [85].

There are several advantages to eliminating anchor boxes[94]:

- It reduces the number of design parameters requiring meticulous adjustment.
- It improves the model's capacity for generalization by removing the hyper-parameters associated with anchor boxes.
- It produces a simpler model, enabling faster training and inference times and lower memory usage by avoiding anchor box-related calculations.

2.1.3.4 Detection-related Classification

As seen in Figure 2.6, both traditional and deep detectors can perform 2D and 3D object detection, depending on detection output.

- 2D object detection: The 2D object detector heavily relies on RGB images to estimate the 2D bounding box of detected objects, guaranteeing accurate results within the image plane. This typically entails using well-defined detection frameworks and datasets.

Regarding the progress in 2D object detection frameworks, they can be separated into anchor-based and anchor-free categories based on the utilization of anchor boxes in the detection process. Most frameworks share a common CNN backbone, enabling an efficient end-to-end detection. However, a recent approach employs the most attractive transformer architecture to support feature extraction and fusion.

However, 2D detection on the image plane lacks crucial information about the 3D position of objects. It fails to fulfill the depth information requirements of real 3D space, hindering its application in trajectory planning and collision avoidance tasks critical for safe driving systems.

- 3D object detection: The field of 3D object detection is gaining more and more attention due to the proliferation of sensor technologies providing 3D information. 3D object detection entails generating three-dimensional (3D) bounding boxes and estimating objects' location, orientation, and size in three-dimensional space. This differs from 2D object detection, which estimates object position and dimensions on the image plane, creating 2D bounding boxes. This method adds a third dimension to dimension regression and localization by introducing depth information in predicted coordinates.

Applications for 3D object detection can be found in autonomous driving, where detectors use sensor data to accurately understand the 3D environment, enabling interaction and navigation in real driving areas. However, 3D object detection is more than just an extension of 2D algorithms, given the following characteristics [116, 115]:

- Precise localization requires depth estimation, but there is a scarcity of labeled 3D datasets.
- Estimating depth becomes harder due to the absence of geometric constraints, particularly for distant or occluded objects.
- Real-time detection becomes difficult due to the increased model complexity and computation burdens caused by the large multi-dimensional data and additional dimension regression.
- Processing diverse sensor data sources requires specific fusion techniques.
- Developing standards for model development necessitates a tailored evaluation system to establish benchmarks.

Frameworks for 3D detection of objects can be classified into broad groups based on input type, i.e., sensor modality. These groupings include image-based methods, point clouds techniques, and multi-sensor fusion approaches [116, 117, 118, 12, 85].

2.1.4 Recent Architectures

This subsection presents popular frameworks proposed for object detection within the deep learning paradigm.

- CNN-based deep object detection frameworks: Due to the deep learning revolution and its promising results, many researchers have recently focused on employing CNNs for object detection tasks. As mentioned in Subsection 2.1.3, CNN-based deep object detection involves two fundamental tasks for accurate detection: object classification, which identifies the category of objects, and object localization, which determines their precise coordinates. As discussed in Subsection 2.1.3.1, CNN-based detection frameworks generally fall into two categories: one-stage and two-stage frameworks, with the most popular ones presented below:

- **One-stage (regression-based) algorithms**

- **YOLO series**: As mentioned in Subsection 2.1.3.1, the YOLO series stands out as an extensively explored framework to build various state-of-the-art meta-architectures for object detection, particularly in real-world applications like autonomous driving. Originating from the work of Redmon et al. [119], YOLO (You Only Look Once) is an anchor-based algorithm known for its speed and accuracy in performing 2D object detection in images and videos. This model simplifies object detection to a single regression problem, where the entire image is analyzed in a single neural network pass to directly predict object bounding boxes and class probabilities, enabling real-time detection. The model uses a CNN for feature extraction, discarding traditional sliding window methods for grid-based image division. Each grid cell predicts a fixed number of bounding boxes and their corresponding class probabilities, using anchor boxes to improve prediction accuracy. The YOLO series includes several versions, ranging from YOLOv1 [119] to YOLOv8 [120], each with improvements in accuracy, speed, and other aspects. This discussion focuses on the YOLOv3 model, emphasizing the rationale behind our investigation of its potential applications.
  - **YOLOv3**: YOLOv3 [121] enhances the efficiency of object detection by merging all necessary stages into one unified network, eliminating traditional steps like region proposal generation and feature resampling for a seamless end-to-end solution. The meta-architecture of the YOLOv3 model, shown in Figure 2.12, consists of a CNN-based backbone network and a dedicated detection head. The backbone is built upon the advanced Darknet-53 architecture, the successor of Darknet-19, with a total of 53 convolutional layers for effective feature extraction, alternating between 3×3 and 1×1 kernels. The model adopts a Feature Pyramid Network (FPN) to support multi-scale prediction, allowing for accurate bounding box estimations at three scales. The dedicated detection head predicts bounding boxes and class probabilities over three distinct layers, each designed to detect objects at varying scales. Instead of the traditional softmax classifiers, YOLOv3 opts for independent logistic classifiers in its output layer. It processes images with a resolution of 416×416, dividing them into $S \times S$ grid cells where each cell is tasked with object detection and can predict up to B bounding boxes. YOLOv3 can generate three feature maps of different sizes (13×13, 26×26, and 52×52), each providing three bounding box predictions per position that include an objectness score, four bounding box coordinates, and class probabilities for C classes. The resulting tensor has a dimension of $N \times N \times [3 \times (4 + 1 + C)]$, where N is the dimension of each feature map (see Figure 2.13). The model predicts class probabilities using binary cross-entropy loss to ensure accuracy. However, it calculates each objectness

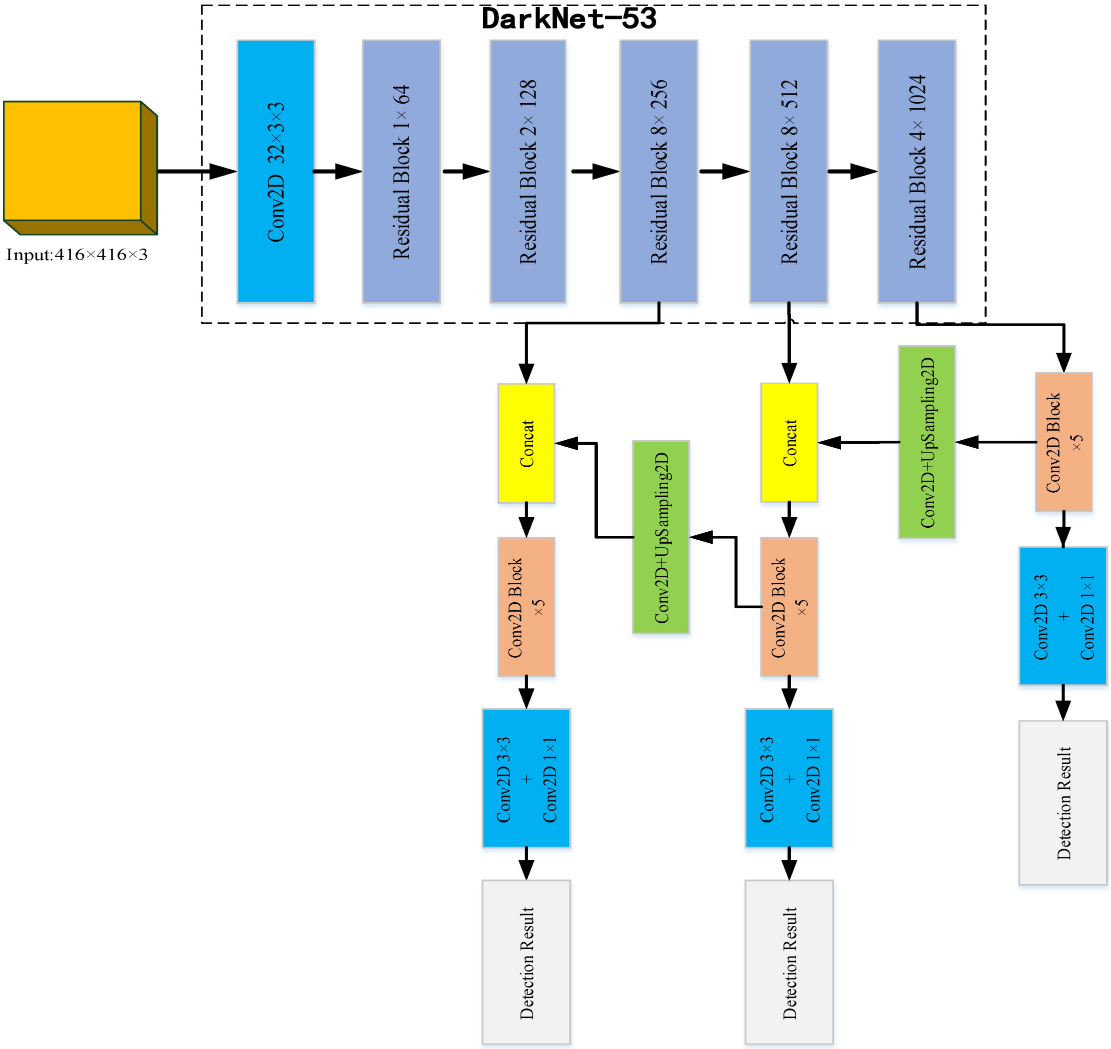


Figure 2.12: Structure of YOLOv3 [1]

score as follows [122, 1]:

$$C_{ji} = P_{i,j}(\text{Object}) \times \text{IOU}_{\text{truth,pred}} \tag{2.1}$$

The objectness score, denoted as C_{ji}, reflects the confidence level of the jth bounding box in the ith grid cell detecting an object. This score is determined by $P_{i,j}(\text{Object})$, a function that evaluates the presence of an object within the bounding box.

The intersection over union ($\text{IOU}_{\text{truth,pred}}$) between the predicted and the ground truth bounding boxes represents an effective metric for evaluating the accuracy of the bounding box predictions. To improve prediction precision, the YOLOv3 algorithm integrates the binary cross-entropy between the predicted and actual (ground truth) objectness scores into its loss function, resulting in more accurate object detection outcomes.

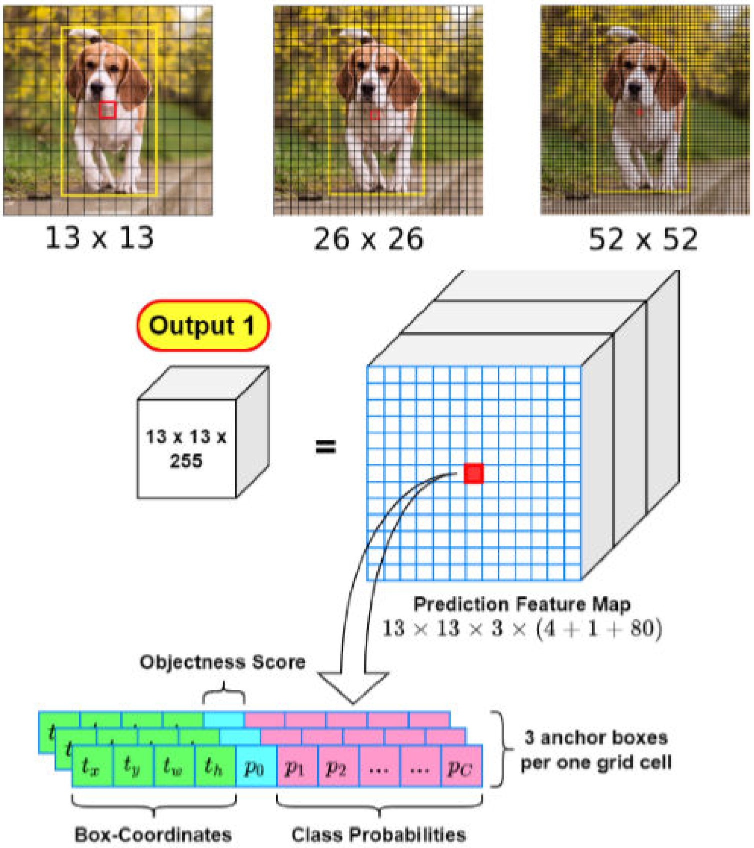


Figure 2.13: prediction output of YOLOv3

$$E_1 = \sum_{i=0}^{S^2}\sum_{j=0}^{B} W_{\text{obj}ij}[\hat{C}_{ji}\log(C_{ji}) - (1-\hat{C}_{ji})\log(1-C_{ji})] \tag{2.2}$$

where S^2 is the number of grid cells in the image, and B is the number of bounding boxes. The terms C_{ji} and $\hat{C}_{ji}$ represent the predicted objectness score and the ground truth objectness score, respectively. Each bounding box position is based on four predictions: t_x, t_y, t_w, t_h, assuming that (c_x, c_y) is the offset of the grid cell from the top left corner of the image. The center position of the final predicted bounding boxes is offset from the top left corner of the image by (b_x, b_y). These values are computed as follows[122, 1]:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \end{aligned} \tag{2.3}$$

where $\sigma()$ is the sigmoid function. The width and height of the predicted bounding box are calculated as follows:

$$\begin{aligned} b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{2.4}$$

The values p_w and p_h represent the width and height of the bounding box prior (anchor box), respectively, which are determined through dimensional clustering. The ground truth box consists of four parameters $(g_x, g_y, g_w, and g_h)$, which correspond to the predicted parameters b_x, b_y, t_w and t_h, respectively. Based on equations 2.3 and 2.4, the true values of $\hat{t}_x, \hat{t}_y, \hat{t}_w$ and $\hat{t}_h$ can be obtained as follows[122, 1]:

$$\begin{aligned} \sigma(\hat{t}_x) &= g_x - c_x \\ \sigma(\hat{t}_y) &= g_y - c_y \\ \hat{t}_w &= \log(g_w/p_w) \\ \hat{t}_h &= \log(g_h/p_h) \end{aligned} \tag{2.5}$$

One component of the loss function in the YOLOv3 model involves calculating the squared error for coordinate predictions. It can be expressed as follows [122, 1]:

$$\begin{aligned} E_2 = &\sum_{i=0}^{S^2}\sum_{j=0}^{B} W_{\mathrm{obj}ij}[(\sigma(t_x)_{ji} - \sigma(\hat{t}_x)_{ji})^2 + (\sigma(t_y)_{ji} - \sigma(\hat{t}_y)_{ji})^2] \\ &+ \sum_{i=0}^{S^2}\sum_{j=0}^{B} W_{\mathrm{obj}ij}[((t_w)_{ji} - (\hat{t}_w)_{ji})^2 + ((t_h)_{ji} - (\hat{t}_h)_{ji})^2] \end{aligned} \tag{2.6}$$

- **Single Shot Multibox Detector (SSD)**: Developed in 2016, the SSD framework outperforms YOLOv1 by applying a multi-scale feature map to object detection. This approach increases the speed of detection while preserving Fast-RCNN accuracy. As shown in Figure 2.14, SSD combines regression concepts from YOLO and Fast-RCNN into a unified model by employing multi-scale regions in various image positions. The algorithm achieves good results even with low input image resolution, with mAP reaching 74.3% and speed reaching 59 FPS. However, it requires manual setting of the default box and has lower detection performance for small objects [123].

- **Two-stage algorithms**
  - **Region-CNN (R-CNN)**: R-CNN represents one of the initial frameworks that features deep learning techniques with conventional methodologies in object detection. This approach incorporates the selective search technique for extracting region proposals and SVMs for classification. R-CNN detection consists of four steps: receiving input images, extracting 2000 region proposals, inputting region proposals, computing CNN features, and using SVM for classification. Figure 2.15 illustrates the main steps of R-CNN [124].
  - **Faster R-CNN**: Developed in 2015, Faster R-CNN leverages the Region

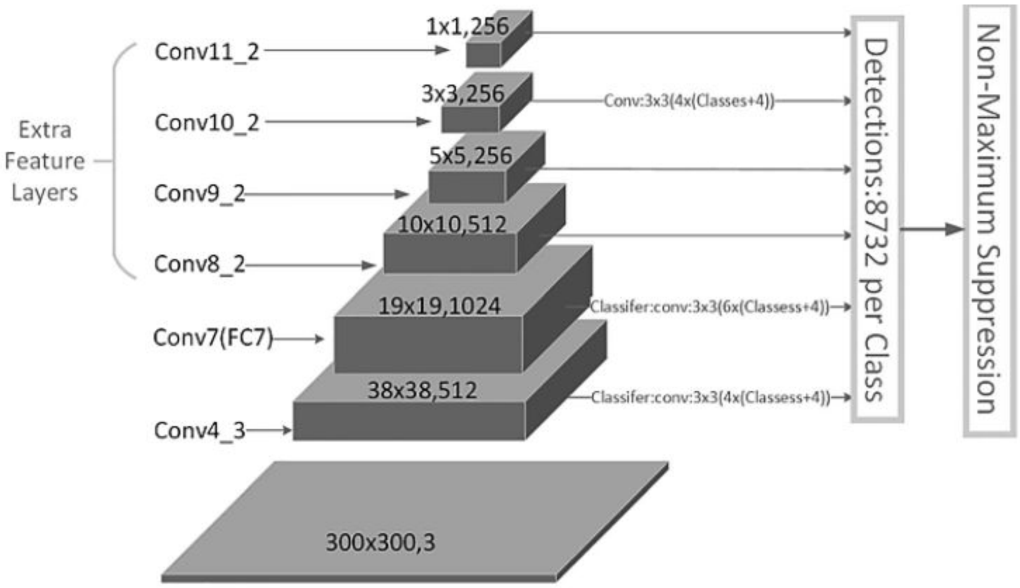


Figure 2.14: Network structure of SSD

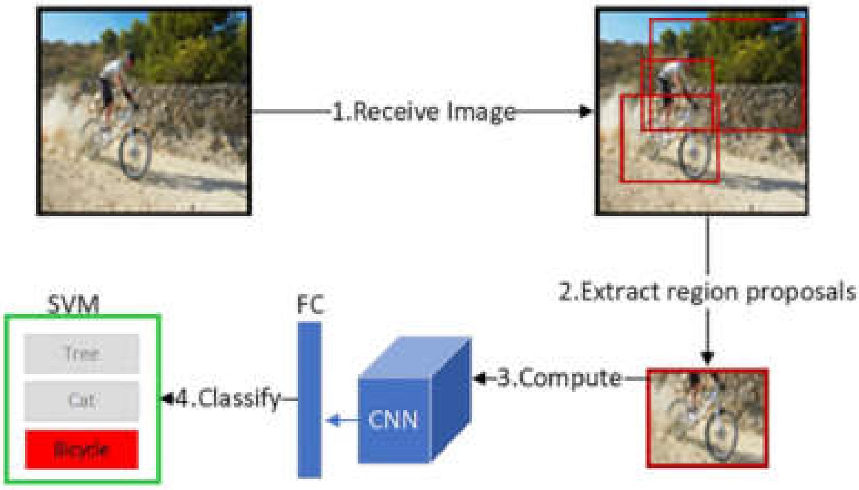


Figure 2.15: R-CNN main steps

Proposal Network (RPN), a CNN-based generator for region proposals and predictor for object boundaries and scores. This eliminates the need for the conventional selective search method, enhancing image processing speed by disregarding the generation of irrelevant region proposals. Despite these improvements, Faster R-CNN still requires significant computational efforts for region proposal classification, making real-time detection unattainable [125]. The detection pipeline of the Faster R-CNN framework is highlighted in Figure 2.16.

– **Spatial Pyramid Pooling Network (SPP-NET)**: SPP-Net introduces a novel approach to address the inefficiencies of feature extraction with fixed-size raw images. The full image is fed into the convolution layer followed by

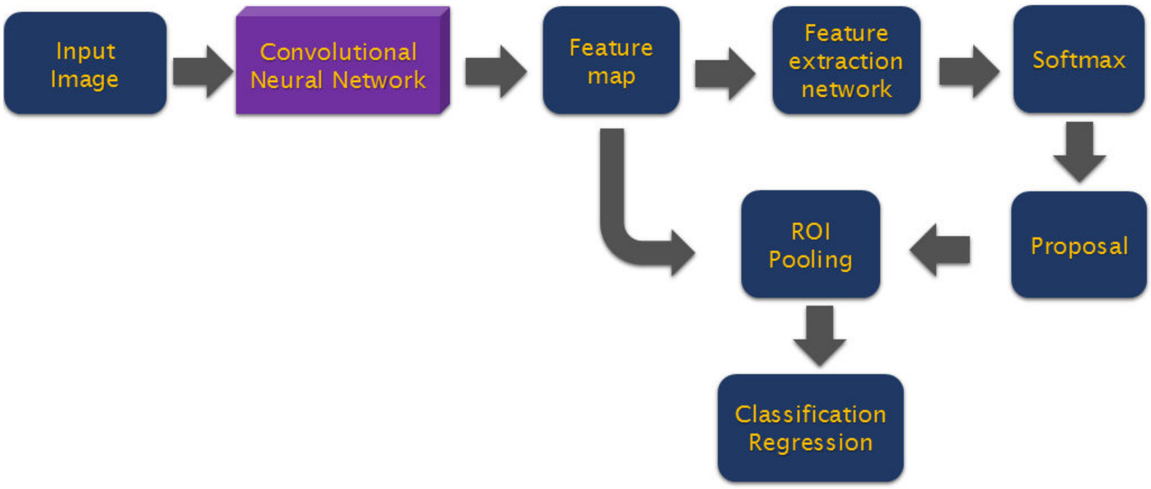

Input Image
Convolutional Neural Network
Feature map
Feature extraction network
Softmax
Proposal
ROI Pooling
Classification Regression

Figure 2.16: Faster R-CNN detection process

processing through the SPP layer, which produces a fixed-length feature map. Maximum pooling is applied to the feature map, creating blocks of varying scales. Despite these improvements, the SPP-Net framework still uses the multi-stage architecture of R-CNN, while proposing new techniques to address certain issues. The key innovation of SPP-Net is its ability to handle images of different scales [126]. Figure 2.17 illustrates the processing pipeline of an input image through SPP-Net.

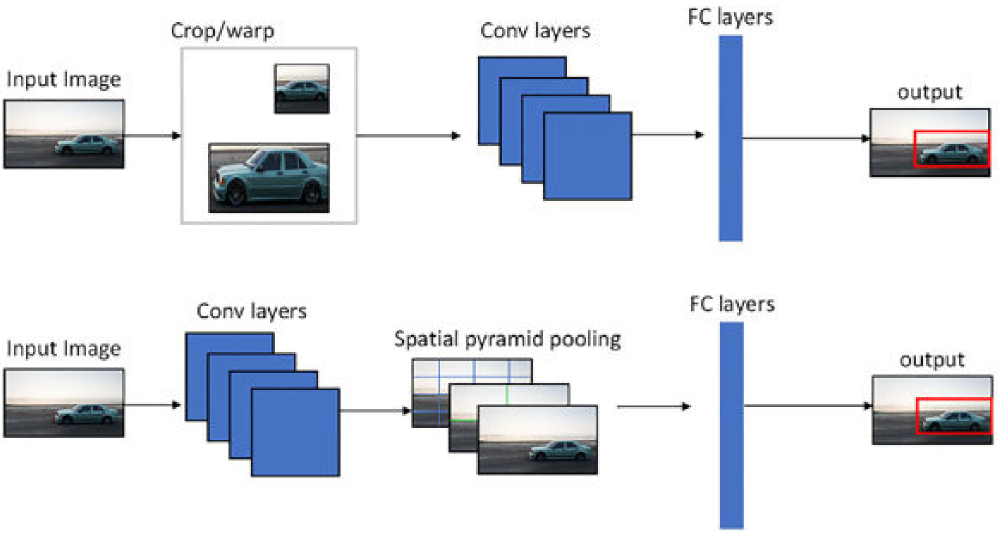


Figure 2.17: SPP-Net input images

Table 2.1 briefly compares several CNN-based models.

-Transformer-based deep object detection frameworks: The adoption of the transformer model for object detection tasks has gained significant prominence. Certain attributes of the transformer have addressed some of the limitations of CNN. As

Table 2.1: Comparison of CNN-based deep object detection algorithms

CNN-based models	Two-stage algorithms	One-stage algorithms	Base model	Limitations
R-CNN	√	×	-Receives input images, -extracts about 2000 region proposals, -inputs region proposals and computes CNN features, -uses SVM to determine classification	-Long training time in multiple stages and feature map calculation, -huge space occupation, -long testing time due to feature map calculation for each candidate region.
SPP-NET	√	×	-Hole image as input into CNN, -uses the SPP-Net layer and accepts image scaling. -SPP layer reproduces image vectors with the same length.	-Fine-tuning parameters difficulty provides low efficiency, -reproduces a new feature map for each image.
Fast R-CNN	√	×	-Uses region of interest pooling Layer instead of convolutional layer, -Multi-task loss for the region proposal and position regression.	-The region proposal extraction is obtained using selective search, -wastes time on region proposal detection.
Faster R-CNN	√	×	-Uses Region Proposal Network (RPN), -Object and score boundaries are predicted at the same time.	-Slow speed,- Nonsufficiency in background learning, -Occluded objects with NON-maximum suppression.
YOLO	×	√	-Combines the generated region proposal and detection in the object detection task, -detects target classification and positioning in the image at one time.	-Low recall rate, -Nonimprovment of detection on tiny objects, -lacks of accuracy.
SSD	×	√	-SSD maintains the accuracy of Fast-RCNN while maintaining the detection speed of YOLO, -SSD puts the regression of the idea of YOLO and the anchor mechanism of Fast-RCNN in one model.	-It's still poor at finding tiny objects, -Manually setting different parameters like the min, and the max size value of the box.
RetinaNet	×	√	-Combins Resnet + FPN, -uses Focal Loss which is a modification of the cross-entropy loss function.	-The problem of noise interference, -provides the necessity of the correct labeling of samples.
CenterNet	×	√	-proposed based on CornerNet, -uses Cascade Corner Pooling.	-The detection of one center point and the two objects as one object

mentioned in Subsection 2.1.3.2, the ViT and DETR series are among the most studied algorithms for object detection based on transformers, and they are described below:

- **Vision Transformer (ViT)**: Vision Transformer [127] is a pure transformer-based detector, initially proposed for the classification task and later tailored to the detection task. It introduces a universal visual backbone that transforms the image into a sequence instead of using convolution for feature extraction and integrates the renowned multiscale feature fusion module [85, 101]. This process starts by dividing the input image into numerous sparse patches, the number of which depends on the dimensions of the input image and the resolution of each patch. For classification, these patches are then linearly embedded and fed into the transformer encoder along with their positional data and a picture representation of the entire image. To adapt the patches for processing across all transformer layers, they are flattened using a constant latent vector and mapped to a dimension size using a trainable projection. Finally, the output related to each patch is fed to a single hidden-layer MLP, which then outputs the predicted class. The model design, illustrated in Figure 2.18, adheres to the original transformer architecture, making it both scalable and efficient. Despite its advancements, the ViT framework has several limitations, such as heavy dependence on data, significant computational demands, single-scale and low-resolution output feature maps, fixed-scale tokens, and challenges in handling long sequences generated by high-resolution images, as well as encoding positional information during the patch-to-vector conversion process. To address these issues, alternative solutions like the Shifted Window Transformer (Swin Transformer) [128] and Pyramid Vision Transformer (PVT) [129] have been proposed [95, 115, 101].

- **DEtection TRansformer (DETR)**: DETR is a transformer neck-based detector that represents a significant shift in object detection techniques by merging transformers and CNNs, moving away from conventional approaches that rely on hand-crafted modules like anchor generation and NMS post-processing. It is a pioneering work that treats object detection as a straightforward set prediction task, allowing for end-to-end training. DETR employs a pre-trained CNN, typically Resnets, for extracting features, fed as a single vector supplemented by positional encodings into a transformer encoder-decoder structure. The novelty of DETR lies in its decoder's ability to process a set of learnable object queries in parallel, directly determining class labels and bounding box parameters. This process avoids traditional object detection complexities by using a bipartite matching algorithm for ground truth labeling and optimizing a Hungarian loss considering both bounding box prediction accuracy and class labels, with SGD minimizing this loss. DETR demonstrates performance comparable to traditional two-stage CNN-based object detectors, such as Faster R-CNN, due to its simpler and more adaptable

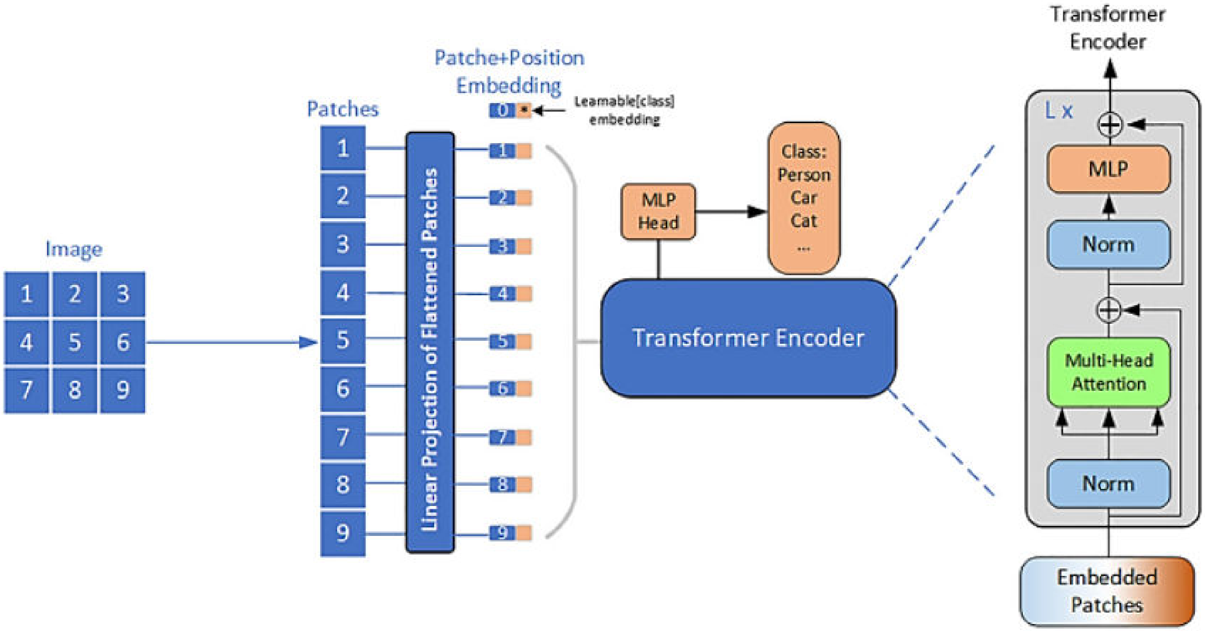


Figure 2.18: Vision Transformer base model structure

architecture reducing reliance on predefined information and specialized layers [95]. However, it faces some obvious shortcomings like delayed training convergence, decreased accuracy for smaller objects detection, efficiency concerns, and a long training period due to substantial data requirements [101, 95]. Following DETR's inception, innovative variants such as Deformable-DETR [130], SMCA-DETR [131], ANCHOR-DETR [132] and YOLOS-DETR [133] emerged to reduce computational needs, particularly regarding the self-attention module. While these enhancements provide notable improvements in various object detection benchmarks, unresolved issues and potential development areas remain, focusing on refining attention mechanisms and enhancing both the number and quality of object queries for greater accuracy and efficiency [134].

Designing a safety-critical AV perception system using a deep learning paradigm requires a comprehensive closed-loop process, including vehicle data collection, valuable data selection and annotation, model training or fine-tuning, validation, and deployment. Building robust deep models for various driving situations necessitates a massive and diverse collection of labeled training samples. Evaluating defects and failures throughout the deep learning development process is essential, as the functional safety risks associated with vehicle operations may raise serious safety concerns.

In response to these challenges, primarily related to passive fully supervised learning paradigms, researchers are seeking ways to reduce the cost of annotation by using less strict supervision techniques. Concurrently, efforts are being made to explore label-efficient and cost-effective learning methods that can be integrated into the deep object detection model training framework, as examined in the following subsection.

2.1.5 Cost Reduction Methods

In practice, the success of most deep object detectors relies on a passive supervised learning paradigm that matches input images to annotations of the objects in those images. However, the dependence on large, accurately annotated datasets, the labor-intensive process of annotating densely distributed road objects, and the high cost of training have prompted the exploration of less supervised methods. This subsection examines the most studied approaches to reducing supervision requirements.

- Training on a small dataset: Acquiring a large amount of accurately annotated data for autonomous driving is expensive and time-consuming. In addition to data augmentation, transfer learning is a commonly used technique for efficiently training models on smaller, sparsely labeled datasets. This technique involves pre-training a network on a large-scale dataset like ImageNet, assuming that extracted features from this large dataset can be shared. The pre-trained network is then applied to the specific task on a smaller dataset. Currently, pre-trained models from the ImageNet challenge dataset, such as AlexNet, VGG, ResNet, Inception, and DenseNet, are widely used for transfer learning.

As depicted in Figure 2.19, there are two primary approaches to applying pre-trained networks to real-world applications: fixed feature extraction and fine-tuning. Figure 2.19 shows that, in fixed feature extraction, the convolutional backbone of the pre-trained network remains unchanged, serving as a baseline feature extractor. Additional machine learning classifiers or new fully connected layers are then built on top of this fixed feature extractor, with training limited to specific datasets. Conversely, fine-tuning involves updating the pre-trained model's weights during training by using backpropagation to adjust all or part of the convolutional baseline for the new task. This method enables the acquisition of task-specific features while retaining the general features learned during pre-training. Fine-tuning can involve adjusting all layers or selectively freezing earlier layers while fine-tuning deeper ones [2].

Alternatively, deep transfer learning (DTL) extends transfer learning by considering learning as a continuous task, e.g., progressive learning. Three primary types of model-based techniques are commonly included in DTL initiatives. Progressive learning entails introducing new layers for training on the target dataset while maintaining some of the pre-trained model's layers unaltered, in contrast to freezing CNN layers or fine-tuning them. Beyond these, there is the adversarial-based technique, which employs relational or adversarial strategies to extract features applicable to both source and target datasets [135].

- Methods based on non-strongly supervised learning: As an alternative to

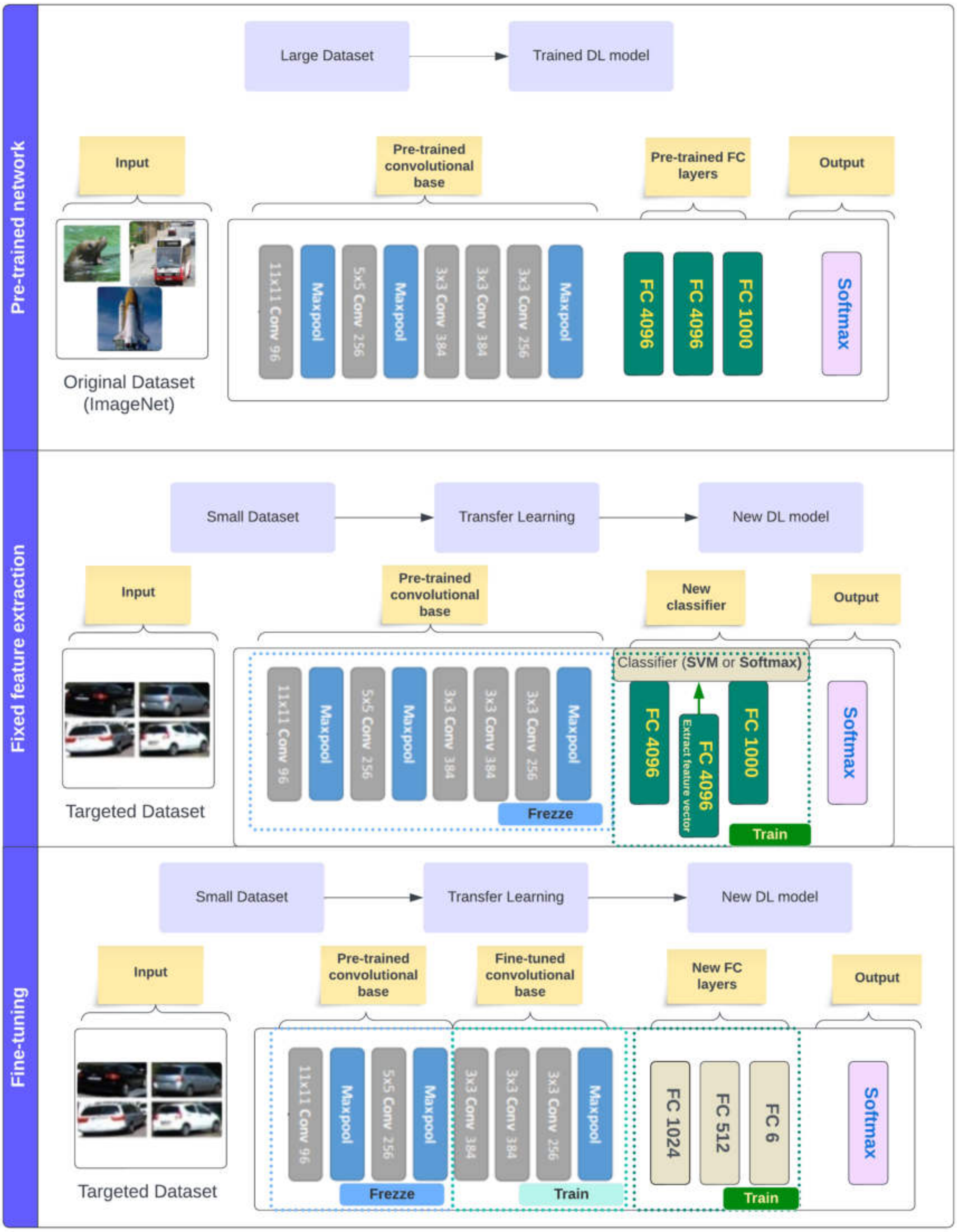


Figure 2.19: Approaches to transfer learning: (a) Fine-tuning. (b) CNN baseline for feature extraction. [2, 3].

transfer learning, a growing number of researchers are interested in exploring the use of unlabeled data to address limited dataset challenges and reduce labeling burdens. Examples of such approaches include semi-supervised learning, weakly supervised learning, and deep reinforcement learning, reviewed below:

- **Methods based on semi-supervised learning**: Semi-supervised learning (SSL) is an effective solution for building safe and robust deep object detector models in dynamic autonomous driving environments. This is achieved by customizing supervised and unsupervised learning algorithms to leverage the large set of unlabeled data alongside a limited amount of labeled data throughout the training framework [136]. This strategy decreases dependence on labeled data and facilitates the exploration of latent patterns in unlabeled examples, significantly reducing the volume of labeled samples required, the learning time of detector models, and the need for extensive labeling effort [137]. Using assumptions such as the smoothness, cluster, and manifold hypotheses, SSL-based object detection algorithms attempt to correlate predictions with learning objectives to determine how similar or nearby samples are likely to be classified. Based on these assumptions, four main strategies currently guide SSL in object detection [138]:

  - **Pseudo labels**: Using model predictions as labels for unlabeled data.
  - **Consistent regularisation**: Ensuring model predictions are stable under small perturbations of the input.
  - **Graph-based**: Leveraging the data structure to propagate labels through the graph.
  - **Transfer learning-based**: Applying knowledge from large datasets to improve performance on the target task.

  Despite the benefits of semi-supervised models, there remains a performance gap compared to fully supervised detectors, especially when dealing with transformer-based models [136].

- **Methods based on weakly supervised learning**: Weakly supervised learning offers an alternative approach to labeling irregular objects or objects obscured by occlusions in high-resolution images or videos, as it only requires coarse-grained, image-level labels. These networks are trained to make decisions on unknown tasks, reducing the complexity of label annotation for real-world driving applications. In the context of object detection, weakly supervised approaches typically utilize certain types of weak supervision, such as using larger-scale but lower-quality training datasets created with cost-effective annotators. In these situations, a few instances are strongly labeled, while other classifiers, heuristic rules, remote supervision, or crowd workers can provide weakly and imprecisely annotated instances [139]. However, the quantity of annotated training data significantly impacts the performance of these detection models. Additionally, the model may struggle to identify rare object classes that have minimal or no representation in the training dataset.

- **Methods based on deep reinforcement learning**: Recent research suggests that object localization can be viewed as a dynamic decision process solvable by reinforcement learning (RL). Inspired by behavioral psychology, RL provides a formal framework for decision-making tasks in various applications, such as autonomous driving. Unlike supervised or unsupervised learning, RL is not dependent on labeled datasets or predetermined rules. Instead, it uses agents that interact with their environment to maximize cumulative rewards and learn optimal policies through trial and error. The agent's learning process is guided by feedback in the form of rewards or penalties, aiming to optimize objectives over time. Unlike supervised learning, RL provides feedback after every action, allowing models to adjust based on experience. RL addresses several important issues, such as delayed input, lack of a supervisor, and sequential decision-making. Deep reinforcement learning (DRL), a combination of RL and deep learning, uses neural networks to manage continuous states or actions. DRL includes both model-based and model-free algorithms [140].

- Enhanced numerical computation: This approach aims to speed up the implementation of object detectors from the bottom, achievable through three methods [141]:

- **Speed up with integral image**: The integral image is a fundamental technique in image processing that enables rapid computation of summations over image regions. It can be applied for faster processing of more generic object features, such as gradient histograms and color histograms. A common application involves computing integral HOG maps to speed up HOG processing, which has been utilized in fast pedestrian detection, despite some loss in accuracy.
- **Speed up in frequency domain**: In CNN-based object detection, convolution is a basic numerical procedure that involves calculating the inner product of the model's weights and the feature map. By leveraging the convolution theorem from signal processing, the Fourier transform provides a method to accelerate convolutions, assuming that the Fourier transform of a convolution operation is equivalent to a point-wise product in the Fourier space of the signals involved. Techniques such as the Fast Fourier Transform (FFT) and the Inverse FFT (IFFT) can expedite this computational process.

- **Vector quantization**: Vector quantization (VQ) is a traditional signal processing technique that approximates a large portion of a data distribution using a limited number of prototypical data points. This method speeds up the object detector by accelerating the inner product process.

- Multi-task, Multi-dataset training: Many existing object detectors are tailored to specific domains and employ a single-task learning paradigm optimized for a singular metric. This approach may overlook other relevant information, leading to limitations in overall model performance and a lack of adaptability across different environments. Multi-task learning involves training a model to perform multiple tasks simultaneously, leveraging shared knowledge and computational resources. By jointly optimizing for multiple tasks across different domains, multi-task object detection models can effectively detect and localize new concepts by leveraging knowledge transfer across related tasks without sacrificing performance on previous tasks (e.g. semantic segmentation and object detection). This facilitates improved decision boundaries, enhanced data efficiency, faster model convergence, reduced overfitting, and increased generalization ability [142].

Moreover, access to extensive datasets featuring bounding box annotations greatly advances object detection. While merging data from various public datasets offering annotations for diverse categories seems advantageous, the differing label spaces pose challenges. Multi-dataset training addresses this issue by leveraging datasets with distinct label spaces to train a unique object detector capable of making predictions across all labels. This method enables the extraction and consolidation of relevant categories from varied datasets, resulting in substantial useful advantages [143, 144].

- Network pruning and quantization: "Network pruning" and "network quantization" are key techniques frequently employed to accelerate CNN models. Network quantizers primarily focus on converting networks to binary, shifting their encoding from floating-point to binary values (e.g., 0/1), thus enabling the use of logical operations to quantify their activations or weights. Modern network pruning techniques often employ an iterative training and pruning procedure, wherein the network architecture is trimmed by repeatedly removing a small subset of weights deemed unnecessary at each training stage [141].

- knowledge distillation: Beyond image classification, knowledge distillation (KD) has been widely applied to object detection tasks. KD is a method aimed at lightweight modeling, transferring knowledge learned from a more complex trained model, *"the teacher,"* to a smaller, lighter model, *"the student,"* performing the same task [85]. The teacher model is initially trained to detect objects. Subsequently, the smaller student model is trained to imitate the teacher model's prediction behavior and achieve performance that is either comparable to or better than the teacher model by picking up essential features guided by the teacher. This method uses less memory and processing power than the teacher model, allowing efficient object detector deployment on resource-constrained devices. Furthermore, knowledge transfer for semi-supervised learning via unlabeled data is facilitated from a fully-supervised teacher model to a student model [136, 85].

The knowledge, distillation algorithm, and teacher-student architecture form the foundation of a KD system. Three main knowledge distillation learning schemes for the concurrent update of teacher-student models: offline distillation, online distillation, and self-distillation [145]. The knowledge distillation learning process involves incorporating the hidden layers' outputs of the previously trained teacher model into the loss functions of the currently learned student model. In contrast, transfer learning involves using acquired parameters from a pre-trained model to initialize the parameters of the model under training [146].

- Detection with domain adaptation: Since most object detection models are trained on independently and identically distributed (i.i.d.) data, handling non-i.i.d. data presents challenges, especially with limited and significantly different image data in the driving domain from the training set. In such scenarios, domain adaptation plays a critical role in bridging the divide between domains.

As a form of transductive transfer learning, domain adaptation addresses the variation between two distinct data distributions: the source domain (abundant labeled data), and the target domain (lacking labels). The primary goal is to mitigate the distribution discrepancy between these domains to ensure that object detection models generalize well in diverse vehicular environments [85, 147].

Several techniques have been investigated to acquire domain-invariant feature representations, including feature regularisation and adversarial training at the image, category, or object levels. Cycle-consistent transformation techniques have also been utilized to bridge the gap between the source and target domains. Furthermore, some methods integrate various techniques to attain enhanced outcomes [141].

- Active learning: Active learning is a machine learning framework that offers a more efficient alternative to traditional passive learning, where the learning algorithm leverages model predictions and interactively queries a human expert (Oracle) for true labels to mitigate the burden of extensive manual annotation and improve performance with fewer labeled instances.

unlike random labeling in the passive learning scheme, active learning algorithms interactively select the most valuable data for labeling from a large pool or stream of unlabeled data. This process is crucial for limited or expensive labels, as it can maximize accuracy while minimizing labeling costs. Common criteria for selecting data include informativeness, which quantifies model uncertainty, and representativeness, which captures input pattern structure. Although no single strategy is universally effective, several heuristics assist in the selection process [139].

Having explored various aspects of object detection, especially within the realm of deep learning, the following section shifts focus to the application of DL-based object detection in the context of autonomous driving. This aims to understand how to achieve reliable and efficient autonomous driving solutions.

2.2 Object Detection for Autonomous Driving

Several attempts to enhance road safety, which motivate the widespread production of AVs, address the perception task as a primary operation towards safe navigation. This is achieved by detecting and localizing road objects, including vulnerable road users. This section covers the intricacies of object detection systems tailored for autonomous vehicles, highlighting the target objects, detection sensors, and datasets related to the proposed algorithms in this context, followed by the challenges faced.

2.2.1 Commonly Detected Objects

In autonomous driving, road objects are categorized into static and dynamic types. Static objects, such as buildings, lane markings, traffic lights, and signs, have a defined shape and fixed, predictable positions. Conversely, dynamic objects, including animals, cyclists, pedestrians, and vehicles, are considered vulnerable to collisions due to their movement.

Detecting dynamic objects is more challenging than detecting static ones, which is relatively straightforward, due to several factors: object occlusion, detection of small objects, and the need to balance accuracy with speed. These challenges are further complicated by issues like sparse visual representation of small targets, complex backgrounds, outdoor environments, and on-board camera calibration errors.

This subsection focuses on detecting the most studied vulnerable road users, who are most likely to collide with the AV, as described below [11, 148]:

- **Pedestrian detection**: Despite the issues related to moving objects, most popular pedestrian detection datasets present a more complicated background and contain many small foreground objects. Additionally, pedestrian detection remains a challenging task due to various pedestrian-specific factors, including large variations, low resolution, and occlusion issues.

    1. **Large variations**: Compared to generic objects, pedestrians exhibit a large variance in scale, which is a critical issue for accurate detection due to the difference in features and appearance between small and large instances. For

instance, small-scale pedestrians are primarily delineated by their contour, while large-scale pedestrians have a more elaborate appearance with facial and body details. As a result, distinguishing small-scale pedestrians from background clutter (e.g., trees) becomes difficult.

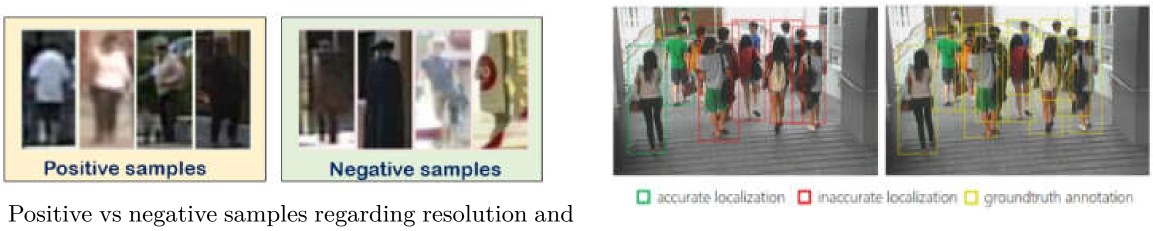
2. **Low resolution and poor visibility**: In low-resolution images, pedestrians are less discriminated from backgrounds. In these cases, discrimination relies more on the semantic contexts. Pedestrians can appear together with cluttered backgrounds, such as traffic signs, pillar boxes, and mannequins in shopping windows, which share similar visual features. Without extra semantic contexts, detectors working with low-resolution inputs cannot discriminate between them, leading to decreased recall and increased false alarms. Pedestrians often appear in low resolution (less than 20×40 pixels) within complex backgrounds, as shown in Figure 2.20a, frequently present as hard negative samples. Moreover, although color cameras have difficulty capturing useful information in low-light conditions, most current pedestrian detectors rely on color images. Consequently, distinguishing between background and foreground under low resolution and poor visibility at night remains a complex problem.
3. **Occlusion**: Detecting highly occluded pedestrians is challenging. In complex scenes, pedestrians gather in groups and are easily obscured by other objects. Depending on the degree of overlap between the occluding object and the pedestrian's pose, occlusion can arise in a wide range of configurations (inter and intra-occlusion). Similar to scale, the detection quality is degraded as the level of occlusion increases. As shown in Figure 2.20b, accurately locating each pedestrian in crowded scenes requires separating feature representations—one representing the visible pedestrian part and the other representing the occluding region. However, detectors often fail to locate each individual accurately, leading to multiple false positives due to inaccurate localization. This problem becomes even worse for CNN-based detectors, where convolution and pooling layers generate high-level semantic activation maps but blur the boundaries between closely-spaced instances.
4. **Illumination variations and other environmental factors**: Detecting pedestrians under low illumination conditions is particularly difficult. Low illumination reduces the amount of information available about edges and other basic low-level features in an image. Additionally, under low-level illumination, image noise becomes more prominent, further degrading performance. Similarly, other environmental factors, such as weather, can also negatively impact detection performance.
5. **Pedestrian modeling and representation**: Compared to general objects, pedestrians exhibit relatively non-rigid orientation and deformation. Typically,

graphical models and part-based models are used, where a pedestrian is represented by templates consisting of components such as body parts, legs, and head. These templates are then used to train classifiers for various occlusion scenarios.

6. **Cross dataset generalization**: Generalization performance is a well-known and fundamental problem in machine learning. Analyzing the generalization performance of a machine learning system involves studying its performance on various test datasets, which may differ from the training datasets.

- **Vehicle detection**: On-road vehicles are generally characterized by Complex backgrounds, varying sizes, models, and orientations, and can be detected using different sensor technologies such as machine vision, millimeter-wave radar, lidar, and multisensor fusion. However, vehicle detection during both day and night vision using optical sensors is very challenging due to the significant within-class variabilities in vehicle appearance. Figure 2.21a illustrates how vehicles may vary in shape, size, and color, while Figure 2.21b shows that a vehicle's appearance depends on its pose and can be affected by nearby objects. In complex outdoor environments, such as those with low illumination conditions (see Figure 2.21c) or at night time, distinguishing between the foreground and background image becomes difficult due to the camouflaging of vehicles. Additionally, the unpredictable interactions between traffic participants and cluttered backgrounds (see Figure 2.21d) add to the challenge. Another key issue is ensuring robustness to vehicle movements in the presence of camera vibration, a significant challenge in vehicle detection and tracking. The primary reason behind the shaking cameras could be environmental factors such as strong winds, resulting in blurred and obscure video footage. Furthermore, long shadows often accompany moving vehicles on the road, reducing the algorithm's ability to detect and classify vehicles during the daytime. To summarize, researchers address three main practical problems in vehicle detection: large variation in lighting, dense occlusion where vehicles overlap with other vehicles, and large variation in scale. Meanwhile, vehicle detection systems should be insensitive to changes in illumination and weather conditions and accurately and efficiently separate vehicles from image sequences. Additionally, vehicle detection systems require faster processing than other applications, as vehicle speed is constrained by the processing rate [6, 149].

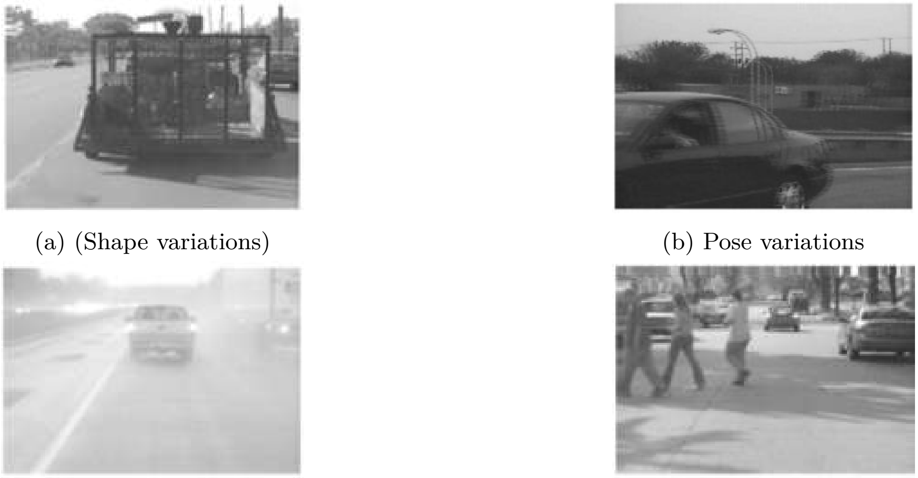
- **Traffic sign detection**: Traffic sign detection and recognition play a vital role in the functionality of AV perception systems. These signs are essential for the safe and efficient navigation of roads, providing a broad spectrum of guidance about road type, prohibitions, speed limits, and height limitations, as illustrated in Figure 2.22a. Traffic signs typically fall into the following categories: warning, prohibitory, and mandatory. For better visibility, they are designed with distinctive colors and shapes. However, precise positioning and categorization are required to accurately detect and recognize traffic signs in vehicle frame inputs. Occlusion, deformation, and long-tailed distribution are some of the issues that greatly impair detector performance in this task (see Figure 2.22b) [7].



(a) Positive vs negative samples regarding resolution and visibility

(b) Detection quality regarding occlusion

Figure 2.20: Illustration of different pedestrian issues [4, 5]



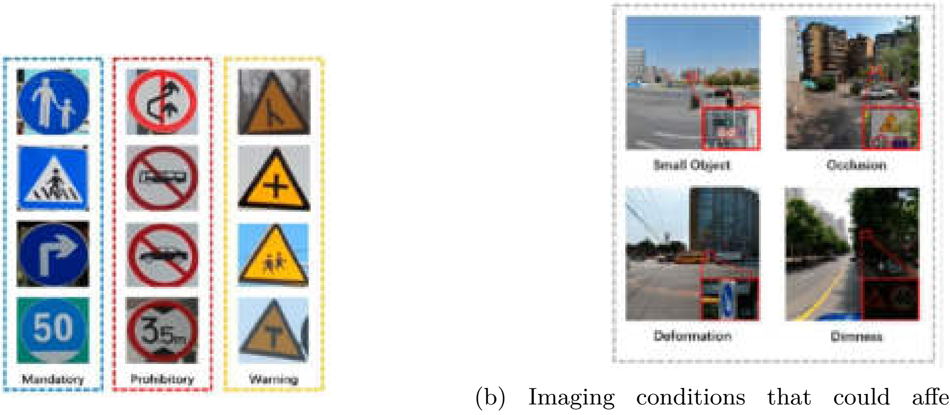
(a) (Shape variations)

(b) Pose variations

(c) Illumination conditions

(d) Cluttered background

Figure 2.21: Some vehicle detection vehicle challenges related to vehicle appearance variations [6]



(a) Traffic signs classification

(b) Imaging conditions that could affect detector performance

Figure 2.22: Analysis of traffic signs [7]

2.2.2 Detection Sensor Technologies

Sensor technologies installed in AVs are among the most crucial sources for providing precise and timely data for ADS, particularly for the scene understanding required by the perception subsystem. As shown in Figure 2.23, the most widely used vision-based sensors for 2D and 3D object detection are outlined in Table 2.2, along with the benefits and drawbacks of each sensor. These exteroceptive sensors can be further classified according to the information they capture, including visual sensors (such as monocular and stereo cameras), LiDARs, radars, ultrasonic sensors, and sensor fusion, as elaborated below[150, 19, 12, 13]:

- **Camera**: Widely used as passive visual sensors in automotive applications, cameras are typically positioned behind the front mirror to capture ambient light and generate a 2D array of colored points (pixels) for AVs. With some modern models incorporating multiple cameras to enhance visibility, they provide critical data for detecting and locating moving or static objects based on captured attributes like color, shape, texture, and depth (via triangulation). However, their performance can be influenced by weather and lighting conditions, which can impact detection accuracy. Against this background, two main vision-based object detection approaches have been investigated: monocular and stereo vision-based approaches. Monocular cameras offer detailed 2D environmental data but lack depth sensing capabilities, whereas stereo cameras use multiple lenses and sensors to extract depth information, though they require more computational resources.

- **Light Detection and Ranging (LiDAR)**: LiDAR is an active time-of-flight (TOF) sensing technology that measures depth data by lighting up the surroundings with laser pulses, capturing the reflected light, and calculating the time each pulse takes to return. To achieve accurate localization and mapping, such as in SLAM (Simultaneous Localization and Mapping) processes, LiDAR creates a point cloud (PCL), a sparse three-dimensional map of the surrounding area. LiDAR's consistent accuracy and reliability across diverse atmospheric conditions are key advantages for its use in AVs. Although LiDAR can withstand changing weather and light conditions compared to cameras, it cannot effectively analyze texture and color characteristics. Despite challenges such as data sparsity and non-uniform point distribution, its depth-measuring accuracy and ability to generate complex 3D environmental maps are invaluable. Given the substantial cost associated with LiDAR technology, efforts have been made to develop less expensive alternatives, such as solid-state and infrared LiDAR, to enable wider application.

- **Radio Detection and Ranging (RADAR)**: Radar leverages radio wave emissions to actively identify near and far road objects by calculating their distance, direction, and speed, using electromagnetic waves across several millimeter wave (MMW) bands, including 24 GHz, 77 GHz, and 79 GHz. The use of higher frequencies improves the system's resolution, allowing for the distinction of multiple objects simultaneously within its visual range. Radar sensors typically function well within short to medium distances (50 - 100 meters), with certain variants capable of object detection up to and beyond 150 meters. They are especially valued in the context of autonomous driving for their robust performance under a variety of environmental settings, serving as a practical and accessible option compared to LiDAR technology. Essential for vehicles equipped with ADAS, radar sensors support functionalities like cruise control and collision detection. Importantly, they can also assess the movement dynamics of detected objects and measure depth information using only the sensing data, similar to LiDAR, aiding in the creation of 3D point cloud maps for navigation and obstacle avoidance.

- **Ultrasonic sensors**: Ultrasonic sensors are designed to actively sense the environment, detect objects on the road, and determine their distance from the AV. By emitting sound waves from sonic transducers at a safe frequency range of 40 kHz to 70 kHz in automotive settings, these devices measure distance based on the ToF principle—the time it takes for the reflected sonic waves to return from an object. Despite their low cost, ultrasonic sensors offer multiple benefits, including effectiveness under challenging weather conditions, long-standing reliability in the automotive industry as parking sensors, and satisfactory accuracy for applications requiring proximity sensing. Nevertheless, their accuracy can be affected by environmental factors like temperature and humidity, which influence sound wave propagation. To deal with this variability, many ultrasonic sensors incorporate algorithms that adjust measurements to reflect current ambient circumstances.

- **Sensor fusion**: Beyond relying on cameras, a vision-based AV perception system requires integrating a supplemental reliable sensing technology like LiDAR, for boosting camera functionality in detecting various on-road object classes under challenging ambient situations. This process, known as "*multi-view fusion*," involves building more powerful detectors by combining information such as point cloud maps and pixel images from different types of sensors. The backup sensor should function reliably, taking over in case of malfunction or failure, thereby ensuring the safety of the AV's occupants.

2.2.3 Datasets for Autonomous driving

This subsection summarizes publicly available autonomous driving datasets for building 2D and 3D on-road object detectors [151, 152, 153, 154].

Table 2.2: Characteristics, advantages and disadvantages of different sensors [11, 12, 13]

	Camera (Monocular/Stereo)	**LiDAR**	**RADAR**
Detection distance	$< 50m$	200-300m	250-1000m
Detection precision	Related to the number of cameras	In the range of 2cm	In the range of 2cm
Detection angle	In the range of 30	In vertical 360; In horizontal 40	In the range from 10 to 70
Detection resolution ratio	< 0.1ř	In vertical 360; In horizontal 40	3-5
Wave length		905nm	1mm-10mm
Advantages	Low cost; different fields of view; high-resolution RGB image; texture; provides longer range and features data; depth calculation; 3D-localization of objects	Accurate depth information; less susceptible to weather and light conditions; panoramic observation	Large field of view; easier to develop, resistant to bad weather; higher accuracy; better resolution; smaller package size.
Limitations	High computational requirements; does not provide straightforward distance calculations; limited by weather and lighting conditions; cannot calculate object velocity	Expensive; lacks texture attributes; no color information	Large package size; shorter sensing range; more data losses; narrow field of view at short distances.

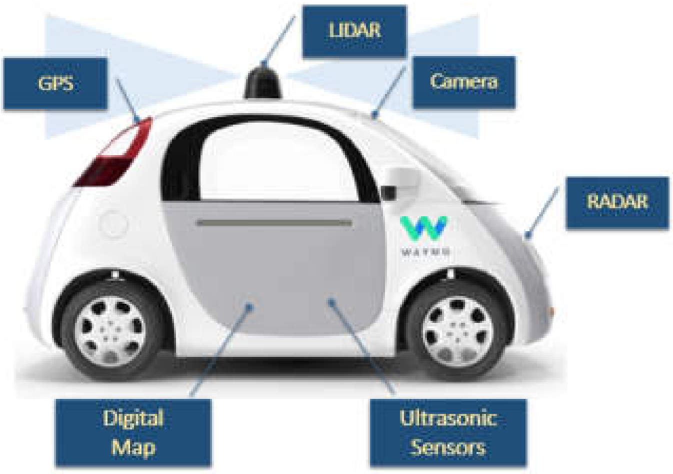


Figure 2.23: Autonomous vehicles sensor.

- **BDD100K dataset**: Released in 2018 by Berkeley DeepDrive Centre, BDD100K [155] is a well-known driving dataset consisting of 100,000 diverse videos, each averaging 40 seconds in length. It includes a series of annotated frames collected under particular weather and time-of-day conditions. It allows research on how weather and illumination affect object detection and tracking, lane detection, semantic segmentation, and multitask learning.

- **Caltech Pedestrian dataset**: Since 2009, the Caltech Pedestrian dataset [156] has become a popular and challenging benchmark for pedestrian detection. It comprises approximately 10 hours of daytime urban traffic video recorded at a resolution of 640x480 and a frame rate of 30Hz. The dataset includes around 250,000 annotated frames, providing temporal relationships between bounding boxes and comprehensive occlusion details, featuring a total of 350,000 bounding boxes and 2,300 distinct pedestrians. To train a pedestrian detector, the dataset is divided into 2975 training images, 500 validation images, and 1575 images for testing. The performance of the detector was initially evaluated using the miss rate vs false positives per window (FPPW) metric, which was further changed to false positives per image (FPPI). The miss rate, calculated in terms of true positives (TP), false positives (FP), and missed detections (FN), is given by the following equation 3.1:

$$Missrate = \frac{FN}{TP + FN} \tag{2.7}$$

  As a benchmark, the commonly used comparative statistic is the miss rate at 1 FPPI.

- **Cityscapes**: Designed specifically for complex urban scenarios, this dataset [157] delivers pixel-level segmentation across 30 distinct object classes, encompassing a variety of vehicles, pedestrians, roads, and traffic signs, making it a crucial benchmark for semantic segmentation projects within urban landscapes. Building upon this, the CityPersons dataset [158] focuses exclusively on person annotations to perform pedestrian detection. The dataset consists of 2,975 training images, 500 validation images, and 1,575 testing images, split from approximately 5000 images captured across several German cities. It includes 35,000 annotated persons, 13,000 ignored areas, and an average of 7 people per image.

- **INRIA**: Derived primarily from holiday shots, the INRIA dataset [159] consists of 2,120 high-resolution pedestrian images in total. Of these, 1,832 images are designated for training, encompassing 614 positive instances and 1,218 negative instances. The remaining 288 images are for general use.

Most of the above-mentioned datasets are primarily used to perform 2D detection of on-road objects based on RGB images captured by cameras. Nonetheless, the following datasets can be useful for performing both 2D and 3D detection using various types of data obtained by fusing RGB cameras and LiDAR measurements [117, 12, 160, 154].

- **Apolloscapes dataset**: The Apolloscapes dataset [161] is a comprehensive training resource for autonomous driving technologies, enabling the building and assessment of modular or end-to-end perception and navigation models in AV. It encompasses a wide range of data types, including images and point clouds, to perform various tasks. Beyond 2D detection, the dataset features frames paired with high-quality annotated point clouds, gathered under diverse driving conditions in Beijing to capture 3D LiDAR-based road objects in challenging traffic patterns involving vehicles, cyclists, and pedestrians.

- **KITTI dataset**: Introduced in 2012 by the Karlsruhe Institute of Technology in Germany and the Toyota Institute of Technology in the United States, The KITTI dataset [162] is a valuable resource for addressing a range of challenging tasks in autonomous driving scenarios. These tasks involve several data modalities, including 2D and 3D object detection and tracking, optical flow, depth estimation, and visual odometry. The dataset offers images reflecting 50 scenes across various driving environments, all captured under sunny conditions. It includes 7,481 images designated as training set and 7,518 images for testing. The training set is split into 3,712 samples for training and 3,769 samples for validation. Approximately 200,000 bounding boxes are annotated, stretching over 15,000 frames. The dataset includes eight classes for labeling, with categories such as car, van, truck, pedestrian, and other types useful for 3D object detection. Despite utilizing only two RGB cameras with 389 pairs of stereo images, the dataset supports both monocular and stereo methods. It is supplemented by optical flow diagrams and point cloud data. For the comparison perspective, the performance evaluation in the KITTI benchmark is conducted using metrics like Intersection over Union (IoU), Average

Precision (AP), and mean Average Precision (mAP). Difficulty levels are categorized as easy, moderate, and hard based on occlusion and truncation criteria. Three variants of AP ($AP_{BBOX}, AP_{BEV}, AP_{3D}$) result from three different IoU definitions: $IoU_{BBOX}, IoU_{BEV}, IoU_{3D}$. These definitions pertain to 2D, bird's eye view (BEV), and 3D scenarios, respectively. The formula for calculating the AP is as follows (Equation 3.2):

$$AP\|_{R_N} = \frac{1}{N} \sum_{r \in R} P_{\text{interpolate}}(r) \tag{2.8}$$

where $P_{\text{interpolate}}(r)$ represents the interpolation function, which is defined as $\max_{r':r' \geq r} P(r')$.This function denotes the highest possible precision corresponding to each recall value larger than or equal to r. The standard Interpolated $AP\|_{R11}$ is used as the primary metric to evaluate the performance in the KITTI benchmark. Furthermore, KITTI introduces the Average Orientation Similarity (AOS). This new metric evaluates the directional alignment between the predicted 3D bounding box and the actual ground truth using cosine similarity to assess their orientation consistency.

- **NuScenes dataset**: Inspired by the KITTI dataset, the NuScenes dataset [163] consists of 1,000 driving scenes, each approximately 20 seconds long. These scenes cover a variety of weather and illumination circumstances and were collected in urban areas. More than 1.4 million camera images and additional metadata are collected using radar, LiDAR, and cameras. The NuScenes dataset is split into 28,130 training frames, 6,019 validation frames, and 6,008 testing frames. Across these 40,000 frames, 23 object classes are manually labeled within 1.4 million bounding boxes. While ten of these classes are used for 3D object detection, the annotation also includes additional information such as range, size, and visibility, in addition to category prediction. The NuScenes benchmark employs a set of seven distinct metrics for evaluation. One of these metrics is the AP based on the 2D center distance on the ground plane. The remaining metrics include Average Translation Error (ATE), Average Scale Error (ASE), Average Orientation Error (AOE), Average Velocity Error (AVE), and Average Attribute Error (AAE). These true positive metrics evaluate translation, scale, orientation, velocity, and attribute errors, respectively. Each class's unique TP metrics are calculated, and their associated averages are denoted through mATE, mASE, mAOE, mAVE, and mAAE. The NuScenes Detection Score (NDS) is estimated using a weighted total of AP and TP measures.

- **Waymo Open dataset**: Released by the company Waymo, the Waymo Open dataset [164] comprises 3,000 driving records, featuring 1,150 sequences containing

600,000 frames captured in urban regions under varied weather and lighting conditions. It includes annotations for approximately 112 million bounding boxes across 200,000 frames divided as 122,200 train frames, 30,407 validation frames, and 40,077 test frames. The dataset also provides around 25 million 3D bounding boxes and 22 million 2D bounding boxes, with annotations available for training and validation only. The evaluation metrics include the Interpolated $AP\|_{R21}$ metric and Average Precision weighted by Heading (APH), where AP is computed over 21 equally spaced recall levels. The Waymo Open dataset distinguishes two levels of difficulty: level 1 for boxes carrying five or more LiDAR signals and level 2 for the remaining non-empty boxes.

2.2.4 Object Detection in Autonomous Driving Environments: State-of-the-Art

The latest advances in deep learning have led to the development of many deep models numerous deep models for detecting both moving and static objects in autonomous driving scenarios, including vehicles, pedestrians, animals, and traffic signs. Broadly speaking, these algorithms fall into two major categories: probabilistic and deterministic object detectors. This subsection examines some of the most notable examples from each category.

- Probabilistic object detection: Unlike traditional deterministic object detectors, probabilistic object detection aims to accurately detect objects while simultaneously estimating the semantic (classification) and spatial (localization) uncertainties associated with each detected object.

By extending frameworks for object detectors, either one-stage or two-stage, the architecture of probabilistic deep object detectors consists of a backbone network, a detection head, and post-processing components. Within the baseline network, techniques like Deep Ensembles or MC-Dropout are commonly employed, which help model epistemic uncertainty in the subsequent detection head. Lastly, in the post-processing stage, standard techniques like NMS, Sample Statistics, Gaussian Mixture, and Bayesian Inference are frequently applied. As a result, probability distributions for object categories and bounding boxes can be predicted. [165]. The following points summarize the main ideas mentioned above:

- Extending well-known deep detector frameworks such as Faster R-CNN, SSD, etc., to output probability distributions instead of fixed values for categories and bounding boxes.
- Utilizing Bayesian neural networks or variational autoencoders to model the inherent uncertainty in object detection problems.

- Employing techniques such as Monte Carlo Dropout, Stochastic Gradient Descent, or other sampling strategies to estimate uncertainty in the context of object detection.

- Deterministic object detection: Most state-of-the-art 2D and 3D object detection methods use CNN and Transformer backbones, as examined below:

- Zhang et al.[166] proposed an image-based real-time 2D object detection method for autonomous driving. The authors combined several popular one-stage object detectors, including YOLOR, which is one of the upgraded versions of YOLOv1. They trained the models separately using various input approaches to improve the detection of each category, especially small objects. They used TensorRT to enhance the efficiency of their detection pipeline during inference for model acceleration. Their proposed detection framework ranked 2nd place in the real-time 2D detection track of the Waymo Open Dataset Challenges, achieving a latency of 45.8ms/frame on an Nvidia Tesla V100 GPU. The study also explored how small objects are statistically distributed within the Waymo Open dataset and how their scale can be enhanced for better detection using different methods like Scale Enhancement, Independent threshold-NMS, and Model Ensemble.
- Han et al. [167] proposed a novel real-time object detection model, Optimized You Only Look Once Version 2 (O-YOLO-v2), specifically designed for detecting tiny vehicle objects in Automatic Driving Systems (ADS) and Driver Assistance Systems (DAS). They enhanced feature extraction by adding convolutional layers at various points in the YOLO-v2 architecture and addressed the gradient vanishing problem with residual modules. By effectively combining high-level and low-level information, O-YOLO-v2 boosted the accuracy of micro object detection and achieved a remarkable 94% accuracy for vehicle detection on the KITTI dataset without compromising speed.
- Bai et al.[168] introduced TransFusion, a robust fusion model that effectively integrates LiDAR and camera inputs for 3D object detection in autonomous driving. The model combines a detection head built on a transformer decoder with convolutional backbones. The first layer utilizes LiDAR point clouds to predict bounding boxes, while the second layer combines object queries with image features. The transformer's attention mechanism enables the model to decide what information should be extracted from the image and where. TransFusion is ranked #1 in the nuScenes tracking scoreboard.
- Feng et al. [169] presented a novel 3D object detector, known as the Structure-Embedding TransFormer (SEFormer). The primary strengths of this architecture lie in its ability to retain and encode structural characteristics derived

from irregular and sparse LiDAR points. Unlike traditional Transformer models, SEFormer learns unique feature transformations for value points while accounting for their distances and relative orientations from the query point. This approach proves more efficient in capturing direction-distance-oriented local structures, which are crucial for the 3D detection of road objects. Extensive experiments on the Waymo Open dataset showed that SEFormer achieves state-of-the-art (SOTA) results, with mAP of 79.02%, surpassing existing works by 1.2%.

- Gupta et al. [170] proposed a CNN-based model for object detection and scene perception in connected and autonomous vehicles. The model focuses on specific image regions, enhancing intelligent adaptive behavior. By integrating a probabilistic attention mechanism that incorporates Transformers, the model accurately identifies critical image areas. Validated on the Berkeley deep drive dataset, the method achieved performance comparable to other deep learning algorithms. The model's performance was evaluated using mAP and speed-accuracy trade-offs.

Moreover, numerous surveys and literature reviews have been conducted to highlight and identify the most commonly used models and architectures for addressing object detection in autonomous vehicles.

- **Single-stage object detectors**: Diwan et al.[171] reviewed the single-stage object detectors, notably YOLO. They discovered that the wide adoption of YOLO's architecture is due to its effective balance of detection accuracy and inference speed compared to two-stage detectors. The comparison was based on both accuracy and speed metrics.
- **Deep generic object detectors**: Liu et al.[5] presented an in-depth review of these detectors. Their findings highlighted the critical factors influencing detection accuracy: the baseline network, detection framework, and access to extensive datasets. To enhance the accuracy, several methods have been explored, including combining different models into an ensemble, leveraging context information in feature learning, and augmenting data. The emergence of PASCAL VOC, ImageNet, and COCO standard benchmarks has facilitated detector comparison.
- **Object detection algorithms**: AMJOUD et al. [115] performed an in-depth analysis of these algorithms, sorting them into three broad types: anchor-based, anchor-free, and transformer-based. Due to its extensive quantity and excellent annotations, the MS-COCO database was the main focus of their investigation; yet, they also employed the Pascal VOC 2007 to evaluate mAP results. The analysis revealed that anchor-based detectors, particularly two-stage ones, yielded the highest mAPs on Pascal VOC 2007, while one-stage anchor-based detectors

also demonstrated considerable effectiveness. Notably, on the MS-COCO dataset, the transformer-based Swin V2-G model alongside the HTC++ backbone emerged prominently, securing mAPs above 50.0%. This success was significantly aided by contributions from ResNets, ResNeXts, Efficient Nets, SpineNet, CSP, and HTC++. In scenarios requiring real-time processing, one-stage anchor-based architectures, especially YOLOv4, were recognized for their optimal balance of speed and precision, outperforming two-stage detectors in this regard. Between 2015 and 2022, transformer-based models, particularly Swin V2-G, led the advancement in the MS-COCO dataset, marking pivotal progress and superior performance in object detection tasks. This progression highlights a trend where single-stage detectors increasingly match the precision of two-stage detectors, with transformer-based designs like Swin-L and Swin V2 pioneering new standards in object detection for vision-based applications.

- **Multi-task environment detection**: Zhou et al.[172] presented a framework for this task. Using ResNet-18 as the backbone, they employed Ultra-Fast-Lane Detection for lane detection and PointNets-based PointPillars for 3D point cloud object detection. The framework consists of three main components: Pillar Feature Net for initial processing, a 2D convolutional backbone ensuring complex feature maps transformation, and a combined output layer for predicting object categories and calculating 3D bounding boxes. By combining both models, they achieved a comprehensive multi-task framework.
- **3D object detection for autonomous vehicles**: Qian et al.[160] conducted a literature review, organizing studies based on data modality: image-based, point cloud-based, and multimodal fusion. Image-based approaches were further divided into result-level and feature-level refinement strategies, where dealing with redundancy and the dependence on supplementary data are key challenges due to the lack of depth information. Point cloud-based methods include voxel-based, point-based, and point-voxel-based, with voxel-based dominating due to its ability to meet autonomous driving application requirements. They also found that point cloud-based methods outperformed multimodal fusion approaches due to semantic heterogeneity between images and point clouds.

2.2.5 Main challenges

The perception of the environment is crucial for AVs, as it enables scene understanding and safe navigation. However, addressing perception errors is essential for the advancement of AVs. Even though DL has demonstrated success in scene understanding, further research is needed in these areas due to the limitations and challenges facing DL approaches for on-road object detection and the performance gap between

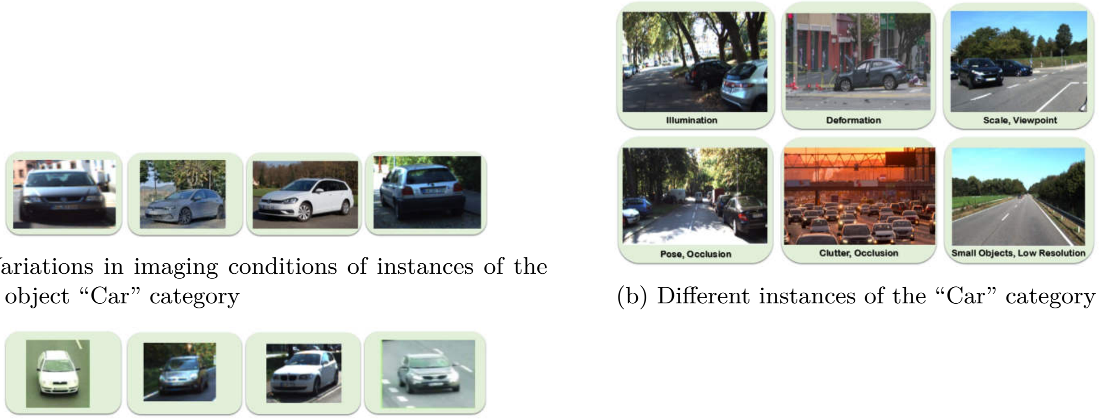
2D and 3D methods. This subsection groups the main challenges as follows [116, 173, 136, 174, 4, 5, 141, 115]:

- Object detection-specific issues:

- **Accuracy-related challenges**: The diverse range of object classes and variations within and between these classes hinder effective road object detection. Intra-class variations can be attributed to intrinsic factors such as color, texture, shape, and size (see Figure 2.24b), as well as imaging conditions like lighting, weather, viewpoint, and occlusion. These variations lead to significant differences in object appearance across different instances of the same category, as depicted in Figure 2.24a. Furthermore, inter-class variations require detectors to discern among tiny distinctions in object categories, necessitating high discriminating power (see Figure 2.24c). Additionally, challenges may arise from digitization artifacts, noise, poor resolution, and filtering distortions. Addressing these challenges often involves collecting a large dataset of properly cropped exemplars for training, which can be costly and requires detectors capable of handling oriented and small objects.
- **Efficiency and scalability related challenges**: Computational complexity and scalability are major concerns for on-road object detection algorithms. The potentially large number of object categories and the wide range of positions and sizes across images increase computational complexity. efficient algorithms are needed to handle the recognition of numerous categories within high-dimensional image representations, particularly for deployment on devices with limited processing resources. Scalability is another challenge, particularly in obtaining training data for detectors that must handle unknown objects, unfamiliar situations, and massive data volumes. Although labeled image samples are useful, obtaining them can be costly, and human annotation becomes unfeasible as the number of images and categories increases. To speed up the learning process, approaches must balance the level of human supervision with the effectiveness of annotation techniques.

- Deep learning-specific issues: As discussed in the previous sections, DL-based object detection frameworks have demonstrated considerable promise in terms of safety, robustness, and efficiency. Notwithstanding their undeniable success, critics have recently highlighted several limitations, as examined below [111, 175, 176, 115, 177, 85]:

- **Sensors limitations**: Sensor limitations are one reason why the perception system malfunctions[116]. Even though sensor fusion is a common approach to reduce the failures of the perception system and address the limitations of multi-modal 2D and 3D object detection, the absence of standardized protocols for incorporating data from various AV sensors such as cameras, LiDAR, radar, and ultrasonic sensors



(a) Variations in imaging conditions of instances of the same object "Car" category

(b) Different instances of the "Car" category

(c) Small inter-class variations

Figure 2.24: Accuracy-related challenges regarding variations in appearance of instances of the same object category [4, 5]

presents a complex issue. This fusion process requires substantial effort to minimize information loss. Additionally, there is a lack of research addressing the fusion of data from ultrasonic sensors, radar, or V2X communication [173, 136].

- **Complex and dynamic scenes & variable weather and lighting conditions**: Real-world driving scenarios are highly dynamic, involving moving objects, varying lighting conditions, occlusions, and complex object interactions. Variations in environmental factors like lighting and weather conditions can degrade image quality, presenting significant challenges. Object detection algorithms must be able to operate effectively in such demanding environments [173]. Furthermore, many autonomous driving datasets primarily focus on typical daytime settings and uniform meteorological circumstances. Despite claiming to have tested several proposals in existing circumstances, further investigation is needed to assess the impact of these conditions on the object detection pipeline while guaranteeing a robust detector across general driving scenarios [116].

- **Generalisation**: In autonomous driving scenarios, both 2D and 3D object detection methods encounter challenges related to poor generalization and insufficient annotations. These issues arise from training models on specific data domains that rely heavily on human intervention and present an imbalance between diversity and labeling. For instance, generalization across motorways, rural areas, and urban driving environments is a significant difficulty due to their distinct characteristics. Moreover, much 3D object detection research has prioritized enhancing benchmark performance over a deep understanding of the performance requirements for reliable driving applications. therefore, research on the safety-accuracy trade-off while maintaining increased contextual awareness and dependability is essential [116]. Beyond the aforementioned generalization concerns, challenges in DL-based object detection include ensuring the accuracy of deep model outputs across diverse scenarios and addressing issues like training fairness to mitigate biases from imbalanced or scarce data. Optimizing DL models for AVs requires advanced techniques due to potential variations in AV reactions [173]. For 3D object detection, focusing on enhancing localization and accurately estimating object parameters is crucial, given the small size of objects and sparsity of point clouds [174]. Additionally, adapting training platforms to handle corner cases, out-of-distribution data, and anomalies is essential [23].

- **Uncertainty**: The inability of most methodologies to provide calibrated confidence estimates for predictions, can lead to risky outcomes in real-world applications. To overcome this issue, further research is needed to identify and quantify uncertainty arising from data, model output, and the behavior of road users. Additionally, strategies for estimating and reducing this uncertainty in detection models that maintain real-time performance must be developed [116].

- **video object detection**: Video-based 2D and 3D object detection for AVs is particularly challenging. However, it offers the advantage of leveraging temporal correlations between consecutive frames to enhance accuracy and enable real-time operations. To improve performance, more efficient and effective methods for motion and intention estimation, along with advanced feature extraction networks, are required [173, 174].

- **Large and open dataset**: In addition to requiring large-scale datasets, many deep object detection models can suffer from overfitting issues (see Figure 2.25) due to limitations in universal benchmark datasets, such as a lack of diversity in captured scenes. To address this, further publicly available datasets encompassing various scenarios, object classes, and modalities are required to develop consistent object detectors for AVs and guarantee resilient functioning across diverse driving scenarios. Creating such large-scale datasets is a challenging but essential task [23, 136].

- **Accuracy-efficiency trade-off**: The trade-off between accuracy and efficiency is a fundamental challenge in on-road object detection due to the small size and imbalance of objects. Nevertheless, achieving these objectives is particularly difficult when dealing with image dimensions and data volumes exceeding those typical of ordinary natural images. While efficient detection methods aim to build high-performance detectors, they often come with a large number of parameters, making them less suitable for deployment on devices with constrained resources. Therefore, developing new strategies that simultaneously enhance accuracy and efficiency is an emerging critical research area.

- **Speed-accuracy trade-off**: Given the computational demands and processing constraints, accelerating the processing of object detection algorithms may compromise their accuracy. Researchers are actively working to enhance both the accuracy and speed of these algorithms by employing more sophisticated architectures and training approaches. This effort is crucial for applications requiring rapid responses and energy efficiency, particularly those navigating complex environments with occlusions or clutter. However, human-designed networks often fail to achieve an optimal balance between detection speed and accuracy [115].

- **Data representations and quality**: Within the field of 3D object detection, transitioning from the consistent pixel distribution in images to the sparse and uneven distribution of point clouds requires the development of specialized models for efficient feature extraction [174]. Additionally, a major bottleneck for training both 2D and 3D object detection algorithms in autonomous driving is the poor quality of benchmark datasets. This issue primarily stems from the methods used to collect these datasets and a lack of variety in the backgrounds. Many current datasets are acquired with cameras mounted on vehicles during regular driving sessions, resulting in a shortage of scenes with crowded settings, occlusions, and overlapping road objects. Consequently, the representation of such complex scenarios is limited. For instance, well-known datasets such as Caltech and KITTI have an average of less than one pedestrian per image, which significantly hampers the efficiency of pedestrian detection algorithms trained with these resources. This highlights the critical need for improvements in data quality to enhance algorithm performance.

- **Label-efficient object detection**: Given the substantial amount of manually annotated road data required for deep object detection algorithms, extracting valuable data from this dataset while balancing the costs and benefits of manual annotation has become a critical challenge for academia and industry. Weakly supervised learning shows promise in addressing this problem by employing less expensive annotations. However, the performance gap between weakly and fully supervised approaches remains significant. Furthermore, semi-supervised and self-supervised learning strategies can concurrently train more resilient 2D and 3D object detection models using a small labeled dataset and a large pool of unlabelled data. Although attempts have been made to tailor self-supervised techniques like contrastive learning to 2D and 3D object detection problems in autonomous driving, the rich semantic content of multi-modal data remains underutilized [174].

- **Robustness for object detection**: 2D and 3D object detectors based on learning methods are often susceptible to adversarial attacks. Perceptual models can be deceived by adversarially introducing perturbations or objects to the sensory inputs, leading to false detections. Addressing this challenge involves developing effective adversarial attack and defense algorithms that are easy to implement and can be applied across different detection models [174].

- **Real-time processing**: In the autonomous driving field, object detection-based collision avoidance systems aim to provide a real-time response to potential crashes by analyzing the video feeds from the AV cameras. However, detecting objects in every video frame can introduce latency issues, as most detectors are trained on image datasets. Model latency can be mitigated by mining frames that are most likely to contain new objects based on correlated spatial and temporal relationships between subsequent frames, which remains an unresolved challenge in this domain [136]. From an edge-cloud perspective, real-time object recognition in self-driving cars can benefit from reducing communication overhead between V2V and V2I servers by designing a deep object detection model with minimal bandwidth requirements [23].

- **Resource constraints**: Efforts to enhance object detector efficiency on physical hardware involve exploring techniques like pruning, quantization, knowledge distillation, and reducing matrix multiplication. Additionally, hardware optimization techniques, like resource allocation and parallel factor adjustments, have been used in conjunction with software co-design to reduce the computational demands and model footprint. While current research has yielded encouraging outcomes, further investigation is required to effectively deal with these issues [136].

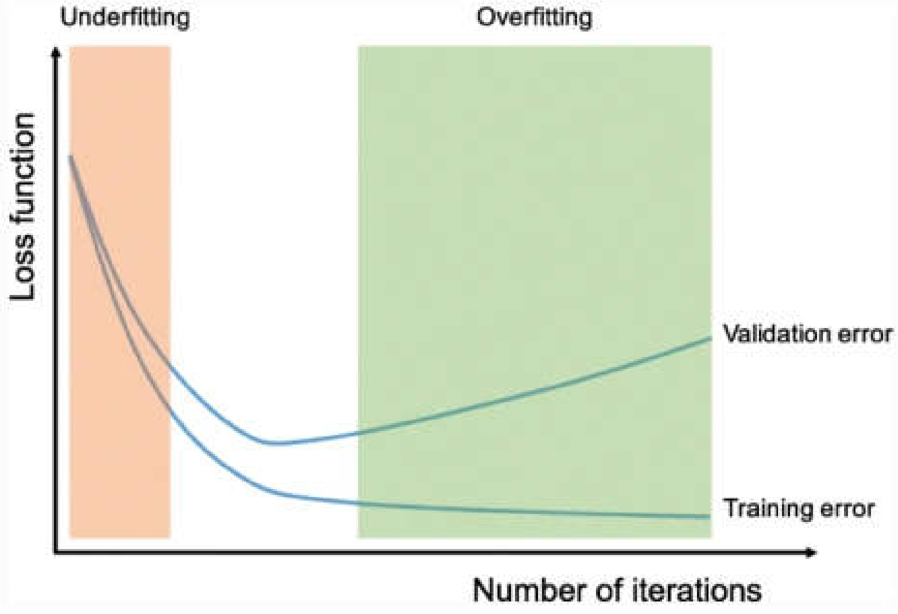


Figure 2.25: Overfitting against Underfitting by monitoring the loss on the training and validation sets during the training iteration [2].

Conclusion

This chapter has provided an overview of the object detection field, with a particular focus on the autonomous driving domain. We began by clarifying key concepts in this field, including principles, categories, popular architectures, and cost-reduction methods. Subsequently, we shifted our focus to object detection methods in the context of autonomous driving. We presented the characteristics of common road objects, the most commonly used sensors, datasets, and evaluation metrics for object detection in autonomous driving. we also introduced the two broad classes of deep object detection models in this context: probabilistic and deterministic object detectors. Additionally, we discussed the shortcomings of representative works within each category. Finlay, we ended by outlining the challenges and issues associated with applying object detection in an autonomous driving context.

In the next chapter, we will explore various concepts related to the active learning paradigm, a prominent approach for reducing costs and addressing the challenges of label-efficient and cost-effective learning in object detection. We will examine different techniques found in the literature, highlighting how cost-effective active learning strategies are tailored for object detection in autonomous driving. We will also discuss the limitations of each approach, to achieve robust and cost-effective autonomous driving solutions.

Chapter 3

Deep Active Learning of Object Detection

Introduction

As discussed in the previous chapter, a significant challenge in applying deep learning for object detection within safety-critical environments, like vehicular environments, comes from passive supervised learning methods' reliance on heavily annotated training data. Object detection tasks particularly require highly accurate annotations, making the manual annotation process both time-consuming and cost-sensitive in the autonomous driving domain. Moreover, the quality of the annotation is subject to collecting and preparing a representative dataset by domain experts.

Active learning emerges as a label-efficient learning scheme that significantly reduces the limitations associated with supervised learning models. This paradigm effectively reduces the need for extensive, high-cost, or challenging-to-obtain annotated datasets by focusing on selecting a minimal yet impactful subset of data for labeling.

This chapter explores the essential principles of active learning, its synergistic relationship with deep learning, and its application in enhancing object detection within autonomous driving. Section 3.1 provides the theoretical background of the field as well as the basis for designing query strategies and how active learning meets object detection. Section 3.2 outlines the innovation of integrating DL with AI and these methods are enhanced. Section 3.3 reviews existing works in the domain of deep active learning for object detection in autonomous driving. Finlay, section 3.4 concluded the chapter by discussion current obstacles and evolving trends in this field. This chapter represents a fundamental part of the literature review, coupled with the previous chapter, which will be used in the following chapter.

3.1 Active Learning

Generally speaking, supervised machine learning models learn from labeled or training data that consists of labeled points. This is denoted by U_{ann} =

$\{(x_1, y_1), (x_2, y_2), \ldots, (x_{n_L}, y_{n_L})\}$, where n_L is the number of labeled points, $x_i \in \mathbb{R}^d$ is the i^{th} data point with d representing the number of features (i.e., the dimensionality of the feature space), and y_i is the label of x_i [178].

Since any supervised learning system must often be "curious" to train on hundreds (even thousands) of labeled instances for delivering better performance, faced with the labeling bottleneck, several attempts have been proposed to overcome this issue. The main approach aiming to perform better with less training based on the key hypothesis of allowing the learning algorithm to choose the data from which it learns. Consider that "random selection" (or random sampling) is the naive passive learning consisting of training the model from a small amount of supervised training samples randomly selected from an unlabeled dataset and labeled by a human. It is a simple scheme with less complexity overhead due to the absence of any interaction with the model's prediction, but comes at the cost of lower accuracy regarding the main constraints of real-world autonomous driving datasets. As an alternative to passive learning, active learning is a label-efficient iterative learning scheme that maximizes model performance under a limited labeling cost/budget, by selecting a small proportion of samples from unlabeled data for labeling and training [8, 9, 179, 180]. This section reviews the foundations and the main concepts of traditional active learning.

3.1.1 Core Principle

In general, active learning (also known as "query learning," or "optimal experimental design" in the statistics literature) is a subfield of machine learning that attempts to improve the effectiveness of ML models constructed with a smaller number of examples and overcome the labeling bottleneck by asking queries in the form of unlabeled instances to be labeled by an oracle. In this way, the active learner is driven by two key ideas: (i) the learner should be allowed to ask questions, and (ii) unlabeled data are often readily available or easily obtained. As a human-in-the-loop machine learning (HITL-ML) approach, active learning is highly label-centered and focuses only on the acquisition of labels based on human feedback emphasizing labels-level human intervention.

Various problem scenarios and settings exist in which a learner can request queries. Regardless of the specific context, active learning can be incorporated into the training pipeline of autonomous driving ML and DL models to maximize the use of valuable data. This process typically follows an iterative learning cycle composed of four key phases: data acquisition, sample selection and annotation, model retraining, and, finally, the evaluation of performance metrics and/or assessment of budget adherence. After the acquisition of unlabeled data, a query algorithm or strategy (sometimes called "utility measures") suggests worthy samples in the unlabeled data and selects informative query instances to be labeled. After that, the model is updated with a small amount of supervised training data labeled by an oracle. Depending on the type of "oracle," which could be an authoritative source, an infallible guide, or a human expert, the recommended training data can be labeled either manually or automatically to assist in the labeling process. Figure 3.1 shows the basic process of traditional active learning.

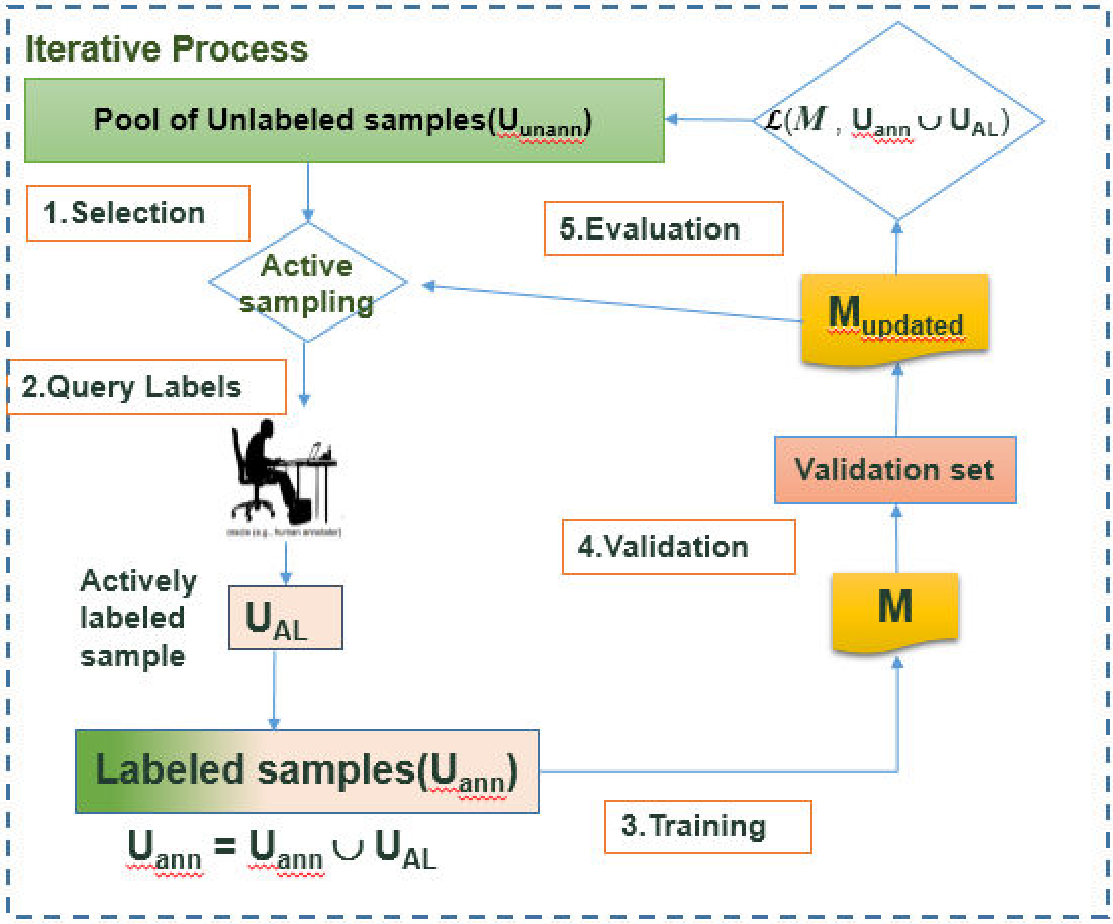


Figure 3.1: traditional iterative active learning cycle.

Further, in most cases, labels come at little or no cost and can be provided free, but for many other more sophisticated supervised learning tasks, labeled instances are very difficult, time-consuming, or expensive to obtain. Against the potential of AL to reduce the amount of instances to be annotated, it would probably not reduce the overall annotation time. Moreover, AL can leverage passive learning in constituting unlabeled data and also be integrated with semi-supervised and unsupervised learning techniques whereas the empirical and theoretical evidence for how and when active learning works in practice can be studied to analyze the active learning process [8, 9, 181, 179]. The main aspects of the AL process, such as scenarios, annotation costs, label noise, are described in more detail below.

- Active learning scenarios: There are several scenarios in which active learners may pose queries. Mainly, three scenarios that have been considered in the literature, where the differences among them is illustrated in Figure 3.2 and discussed below [8, 9]:

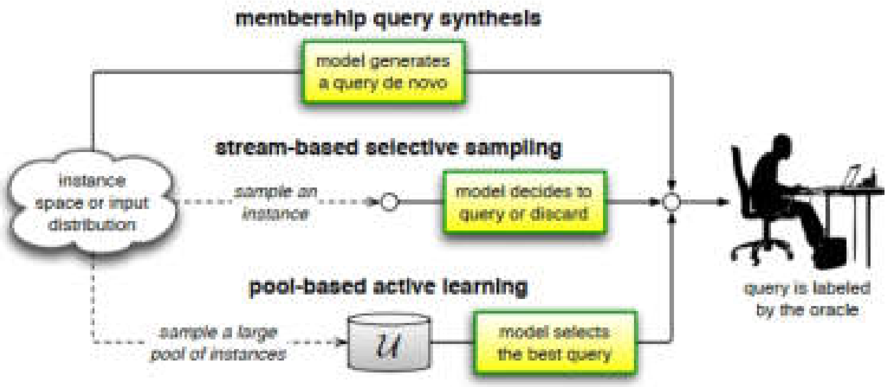

membership query synthesis
model generates a query de novo
stream-based selective sampling
instance space or input distribution
sample an instance
model decides to query or discard
pool-based active learning
sample a large pool of instances
$\mathcal{U}$
model selects the best query
query is labeled by the oracle

Figure 3.2: main active learning scenarios difference illustration [8].

1. **membership query synthesis**: An early and popular scenario of active learning is *membership query learning*, in which learner may query for labels for any instance in the input space that is not yet labelled. This involves generate queries from scratch, rather than relying on natural distribution samples. This strategy is particularly practical and effective in finite domains and has been adapted to regression tasks. While query synthesis is applicable and beneficial for various problems, it may pose challenges when human annotators are involved. However, in fields where labels are derived from experiments rather than human annotation, query synthesis offers valuable opportunities for advancing automated scientific discovery.

2. **stream-based selective sampling**: *Selective sampling* offers a practical alternative to query synthesis in active learning. Under this approach, the assumption is that acquiring an unlabeled instance is inexpensive or free, allowing it to be sampled directly from the true distribution before deciding whether to label it. This method is often referred to as *stream-based or sequential active learning*, where unlabeled instances are evaluated one by one, and the decision to query or dismiss each is made on the spot. Selective sampling can resemble membership query learning in scenarios where the input distribution is uniform. The decision to query an instance in selective sampling can be approached in several ways. One method involves using an informativeness measure or query strategy to make a biased random choice, with more informative instances being more likely to be selected for querying. Another strategy focuses on identifying a region of uncertainty within the instance space, querying only those instances that reside within this ambiguous area. A more structured approach defines a version space, which encompasses the subset of the model class that remains consistent with the current labeled dataset, and queries are made on instances that lie within this yet-to-be-explored region.

3. **pool-based active learning**: Pool-based active learning is particularly suitable for scenarios where large sets of unlabeled data are readily available (possibly collected

using passive learning techniques), as often found in tasks like object detection. This approach starts with a small labeled dataset U_{ann} and a much larger pool of unlabeled data U_{unann}. The strategy involves selectively querying instances from this pool, which is generally considered to be static. The selection of queries is typically done in a *greedy manner*, based on some measure of informativeness to assess the potential value of each instance in the pool, or a subset thereof. After a query is picked and its label obtained, it's added to the labeled set U_{ann}, and the learning model is updated or retrained with this new information. Then, another query is selected from the pool, considering all previously labeled instances, and the cycle continues. This method allows for the efficient expansion of the labeled dataset by focusing resources on labeling the most informative instances, thereby improving the learning model's performance with fewer labeled examples. The key distinction between pool-based and stream-based active learning lies in their approach to data selection. Stream-based active learning processes data sequentially, making immediate decisions about whether to query each instance as it arrives. This approach is particularly useful in situations with limited memory or computational resources, such as with mobile or embedded systems. On the other hand, pool-based learning assesses the entire dataset or a significant portion of it before selecting the most informative instances for querying, which can be more computationally intensive but allows for a more informed selection process. This makes pool-based active learning a preferred method in many real-world applications where computational resources are not as constrained, and where the most efficient use of labeling resources is a priority.

- Batch-Mode active learning: As illustrated in Figure 3.3, in most active learning settings, queries are selected in serial, i.e., one at a time. The learner typically examines a large pool of unlabeled data and then selects a single instance (the most informative one) for label querying using a one-by-one, single-criterion query. This is not true in more practical situations where the serial selection of queries can lead to inefficiencies in terms of labeling resources use, such as the time of human annotators. This inefficiency comes from the delay restriction, as the next label cannot be queried unless the model is updated beforehand.

It may be more practical to acquire labels for multiple instances simultaneously. Batch-mode active learning enables the learner to query a group of instances (batch), suitable for distributed and parallel labeling environments. This approach is also beneficial for methods or models that are expensive or data-intensive and have slow training processes, such as deep learning, which necessitate selecting a batch of many instances at each iteration. In such settings, the main idea is to select a set of queries $\mathcal{Q} \in U_{unann}$ to be labeled concurrently with model re-training, or in parallel if supported by the experiment or annotation environment. A straightforward approach to creating this batch is to simply use a myopic strategy by evaluating all potential query instances and selecting the "Q-best" ranked instances based on some utility measure as a metric. Unfortunately, this does not work well since it fails to consider both diverse and informative "best" instances. Instead, the instances in $\mathcal{Q}$ need to be both informative and non-redundant to optimize the use of labeling resources. To achieve this, the query strategy explicitly incorporates multiple criteria to evaluate both informativeness and representativeness, ensuring the selection of instances that are both informative and representative for querying.

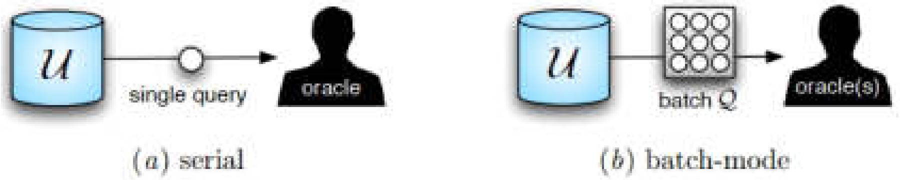


Figure 3.3: Serial vs. batch-mode active learning [9].

- Active learning with costs: Since some acquired data may be noisy and contain no useful features that are relevant in most real-world applications, it's crucial to note that not all data are equally important for model construction from the perspective of labeling. Additionally, in some learning problems, not only the cost of obtaining labeled data can vary from one instance to another, but also the label quality. In such a context, the main goal of active learning is to minimize the overall cost of training an accurate model by reducing the dependency on a huge amount of labeled data. Nonetheless, reducing the number of labeled instances does not necessarily guarantee a reduction in overall labeling cost.

Undoubtedly, given the differential value of different data and the trade-offs between accuracy and cost, an important question that arises in active learning is how to approach cost-sensitive and cost-effective active learning regardless of whether annotation costs are assumed to be fixed and known to the learner before querying or variable and not known, e.g., when the labeling cost is a function of elapsed annotation time [8, 9].

- Noisy Oracles: From a label-centered perspective, active learning is a framework that significantly relies on human feedback to acquire labels for improving the effectiveness of ML models with a strong focus on the acquisition of labels. Within this framework, a key assumption in most active learning research is that the quality of labeled data from the oracle is high. In practice, if labels come from an empirical experiment like biology, chemistry, or clinical studies, it is reasonable to anticipate some degree of noise to result from the instrumentation or experimental setting. Even if labels come from human experts, their reliability is not absolute because: (i) some instances are implicitly difficult for both humans and machines, and (ii) human annotators are prone to distraction or fatigue, leading to inconsistencies in the quality of their label assignments. [9].

3.1.2 Query Strategies

The primary focus of all active learning scenarios is how to determine the "valuability" of unlabeled instances, whether generated anew or sampled from a specific distribution. By establishing criteria for assessment, the query strategy (also known as the acquisition function or selection strategy) determines if a single sample (or a group of samples) merits consideration as a candidate for labeling by the annotator, based on predefined criteria and rating scores. Throughout the literature, these query strategies have been classified in terms of several factors, including the number of samples asked for labeling at once, as follows [180, 178]:

- **One-by-one query**: Known as a single-instance strategy in which querying only one sample at a time leads to updating the underlying learning model whenever a new sample is introduced. Furthermore, adding just one annotated instance is unlikely to boost the performance of learning large-scale models in the context of deep learning, leading to a complex, laborious, and time-consuming process.
- **Batch query**: This strategy is particularly well-suited for selecting multiple samples at once, especially in the case of expensive/data-hungry methods like deep learning. Typically, to pick a batch of $\mathcal{K}$ most informative samples, the straightforward approach is to iterate a single-instance query for $\mathcal{K}$ times. Meanwhile, the diversity of samples and the amount of information each provides should be investigated as criteria during the selection process to prevent picking similar instances within the batch.

On the other hand, query strategies are divided into the following categories based on the quantity of information at hand [178]:

- **Data-based**: Strategies involving low-level knowledge, limited to raw data and its labels, fall into this category.
- **Model-based**: With a focus on determining the uncertainty within a class, strategies in this category make use of information about the data and the model but do not take predictions into account. In the case of the well-known expected model change strategy, which fall into this category, the updated model, using the previously labeled instances, queries a new unlabeled instance that has expected to clearly change the model parameters.
- **Prediction-based**: These strategies, such as uncertainty sampling, leverage all available knowledge, including data, models, and predictions. They target the uncertainty between different classes, where the most uncertain unlabeled instance is queried based on model predictions.

- **Agnostic strategies**: These methods operate only on the information drawn from the unlabeled data pool, discarding any knowledge generated by the trained model and any assumptions about the accuracy of its decision boundaries.
- **Nonagnostic strategies**: To pick and query new unlabeled instances, this method mostly relies upon the trained model.

To select a new candidate sample for querying, the query strategy assesses the instances in U_{unann} and produces utility scores/values using a utility function ($\sqcap$). These scores determine which sample (or samples) is likely to be considered for label querying with regard to the selection criteria, namely informativeness, representativeness, and hybrid. This subsection provides more details.

- **Information-based query strategies**

Within this category, the query strategy's selection process revolves around measuring the informativeness criterion. For this purpose, the utility function is designed to ascertain scores, aiming to identify the most informative samples that are likely to be approximately the decision boundaries. Numerous instances of this approach exist within this category, namely *heterogeneity-Based* and *Performance-Based Models* as discussed below [181, 178]:

Heterogeneity-Based Models: These frameworks intend to identify the areas characterised by the highest heterogeneity, whether it be in regard to classification uncertainty, dissimilarity with the present model, or disagreement amongst a committee of classifiers.

- **Uncertainty Sampling**: Based on the uncertainty heuristic, the uncertainty sampling strategy aims to label instances where it is least certain about the correct label (or most uncertain). For overcoming the binary classification scenario, the simplest approach involves applying probabilistic learning algorithms on an instance and requesting its label if the predicted probability of the most probable class is close to 0.5. However, normalising the classifier's estimated probabilities is necessary. This strategy is also known as the least confident (LC) approach. For multi-class problems, the general formula is as follows (Equation 3.1):

$$x^* = \arg \max_{x \in U_{unann}} (1 - P_h(\hat{y}|x)) \tag{3.1}$$

where x^* is the least confident instance, $\hat{y} = \arg\max_y P_h(y|x)$ is the class label of x with the highest posterior probability using the model h, and $P_h(y|x)$ is the conditional class probability of the class y given the unlabeled point x.

Margin sampling, on the other hand, takes into account the information associated with the remaining classes in the distribution that has been overlooked. Using this technique, the difference between the probabilities of the first and second most likely class labels is computed as follows (Equation 3.2):

$$x^* = \arg\min_{x \in U_{unann}} \left(P_h(\hat{y}_1|x) - P_h(\hat{y}_2|x)\right) \tag{3.2}$$

where y_i ranges over all possible class labels, and $P_h(y_i|x)$ is the conditional class probability of the class y_i for the given unlabeled point x. The instance with the largest entropy value is queried. This means that the learner queries the instance for which the model has the highest output variance in its prediction.

- **Query by Committee**: In the query-by-committee (QBC) approach, a set of models (or committee members) $\mathcal{H} = \{h_1, h_2, \ldots, h_g\}$ is trained on different subsets of samples drawn from $\mathcal{U}_{unann}$. After that, the disagreement between these committee members is estimated, and then the most informative points are queried where the disagreement between the committee members is the largest.

  There are numerous ways to quantify the degree of disagreement among committee members. The vote entropy approach is one such method, and it works as follows (Equation 3.3):

  $$x^* = \arg\max_x - \sum_i \frac{V(y_i)}{m} \log \frac{V(y_i)}{m}, \tag{3.3}$$

  where y_i denotes all potential labels, m signifies the number of members in the committee, and $V(y_i)$ stands for the count of votes a label garners based on predictions from all classifiers.

- **Expected Model Change**: This approach focuses on selecting data points for querying that are anticipated to cause the most significant alterations to the current model. Specifically, it targets those points believed to have the most substantial influence on the model, without concern for the labels these queries might yield. This strategy is particularly relevant to models that employ gradient-based optimization methods, such as certain probabilistic models that are discriminative in nature. The expected change in the model, due to querying an instance X with an unknown label, is quantified by calculating the expected change in the model's gradients, denoted $\delta g_i(X)$, for a hypothetical label i, and integrating this with the posterior probability p_i of the instance being assigned label i, given the current labeled data. The definition of the expected model change $C(X)$ about the instance X is as follows (Equation 3.4):

  $$C(X) = \sum_{i=1}^{k} p_i \cdot \delta g_i(\overline{X}). \tag{3.4}$$

The instance $\overline{X}$ with the largest value of $C(\overline{X})$ is queried for the label. This approach, though, requires significant computational resources, particularly for issues featuring high dimensionality or extensive labeled datasets. Moreover, failing to scale the features results in a significant reduction in its effectiveness.

Performance-Based Models: Occasionally, heterogeneity-based models may discover noisy and unrepresentative portions of the data due to their attempt to find the most unknown regions of the space (on the basis of the current labeling). Of fact, the specific effects of employing such a strategy depend greatly on the data. There are two classes of techniques that are based on the performance of a classifier on the remaining unlabeled instances.

- **Expected Error/Prediction Change**: Using the remaining unlabeled dataset ($\mathcal{U}_{unann}$), active learners predict the future error of the model they learned using $\mathcal{U}_{ann} \bigcup \langle x^*, y^* \rangle$. Subsequently, they then seek to query instances that are likely to decrease the anticipated future error, such as by aiming to minimize the expected 0/1-loss, as shown below (Equation 3.5):

$$x^*_{0/1} = \arg\max_x \sum_i P_h(y_i|x) \left(\sum_{j=1}^{n_U} 1 - P_{h+\langle x^*, y^* \rangle}(\hat{y}|x^{(j)}) \right) \tag{3.5}$$

  where $P_{h+\langle x^*, y^* \rangle}$ is the new model after retraining it with $\mathcal{U}_{ann} \bigcup \langle x^*, y^* \rangle$. Therefore, a validation set is required in this category to evaluate the performance of the learned hypotheses. Seen as an offshoot of the expected error reduction strategy, The variance reduction method is a further variation of this approach where active learners focus on selecting instances that lead to a decrease in the model's variance. This action, in turn, helps to lower the model's future generalization error.

- **Representation-based query strategies**

By leveraging the organization of the unlabeled data, the query strategy attempts to identify specific samples that capture the overall structure of the entire input space. Consequently, To find the most representative samples in $\mathcal{U}_{unann}$, the utility function in this category assesses how representative each sample is. Because it targets points in densely populated regions, the representation-based method outperforms the information-based strategy in terms of exploration capabilities. Below is a description of some methods used in the representation-based query strategy [181, 178]:

- **Density-based approach**: This strategy obtained representative points by sampling instances from dense regions within the input space. Similarity-based methods are the most common in this context, which involve measurements like the distance between feature vectors. For instance, cosine similarity, KL divergence, and Gaussian similarity have been well-studied.

- **Cluster-based approach**: Cluster centers are utilized in clustering techniques to pick representative points. The closest neighbors to the cluster centers are chosen in this case after clustering the entire input space. The effectiveness of this approach ultimately relies on the clustering algorithm and the parameters that are selected.
- **Diversity-based approach**: To address the issue of redundancy in selected points, the query picks unlabeled data with greater diversity compared to already labeled ones. To evaluate the diversity, the angles between the unlabeled data and all the labeled datasets $\mathcal{U}_{ann}$ are calculated. If unlabeled data differs considerably from the others in $\mathcal{U}_{ann}$, then it is picked for querying. However, it's important to combine this strategy with others for better efficiency to prevent querying outliers.
- **Hybrid models**

Usually, either informative or representative unlabelled instances are selected in the commonly used single-instance, single-criterion active learning approaches. Nonetheless, several studies have explored combining various criteria for query selection, ensuring that the selected instances will display the following characteristics:

1. **Informative**: The selected instance for querying will either be near the decision boundary of the learning model, reflecting criteria such as uncertainty, or it will be positioned far from the currently labeled instances, thereby introducing new knowledge into the feature space.
2. **Representative**: Beyond representing a cluster of other unlabeled data, the picked instance for querying ought to exhibit a lower likelihood of being an outlier. Some research methodologies, for instance, use a min-max framework to score the informativeness and representativeness of the unlabeled instances to perform the query selection process.

3.1.3 Active Learning Meeting Object Detection

While active learning has has demonstrated success in image classification tasks, further research is needed to fully explore its applicability to other tasks, such as object detection. By picking the most informative samples from the unlabeled dataset according to a set of criteria, active learning for object detection targets to minimize labeling costs while boosting training efficiency for more effective detection models. This approach enables achieving satisfactory performance with fewer labeled examples, raising the question of how to design suitable metrics and query strategies tailored for object detection algorithms.

In practical terms, object detection encompasses both classification and regression tasks, with the detection model providing predictions for both classification (identifying object classes) and regression (predicting bounding box coordinates) as explained below:

- *Classification*: This component is tasked with determining object instances present in a given image or a specific part of the image. In object detection models, classification predicts an object's category, which could be a vulnerable road user. As part of the prediction, each class is given a probability score that represents the model's confidence in the presence of that particular class.

- *Regression*: The regression component focuses on the localization (spatial location) of the object within the image. It predicts a set of bounding boxes, which are rectangular coordinates that outline the location and size of the object. The regression task involves determining the x and y coordinates of the box's corners or the center, along with its width and height.

In addition, post-processing techniques, such as NMS, can be also used.

Given the significant progress in active learning for classification tasks, the application of active learning to regression tasks has received much less attention. Moreover, an additional complexity to active learning strategies may be introduced when dealing with object detection due to the potential presence of multiple objects within a single image. At this point, a straightforward strategy might involve applying standard baseline techniques to each prediction and then employing aggregation methods to derive heuristics for a more comprehensive acquisition function. In light of this, methods for calculating and estimating rating scores at several levels, such as pixel, image, box, or by combining the aforementioned elements, are crucial.

The predominant DAL frameworks tailored for object detection rely on query strategies that evaluate classification uncertainty as a confidence metric, typically by leveraging the softmax layer prediction. Moreover, methods like Monte-Carlo dropout or stochastic regularization are utilized to produce class outputs, facilitating variational inference implementation and posterior distribution computation of network predictions. Alternatively, multiple layers' output from the detector network's backbone can be used to build a committee of classifiers, and the degree of disagreement between their predictions indicates how informative the framework is.

However, there are certain difficulties associated with employing conventional AL algorithms in the context of deep object detection. Below are few of them [182, 183, 184]:

- Using the softmax layer to determine the label probability distribution might lead to inferior performance than random sampling. This is associated with the unreliability of the final output's softmax predictions, rendering it unsuitable for uncertainty heuristic utilization.

- Developing sampling techniques for object detection is more challenging than developing them for classification tasks. This is due to the presence of background elements, which significantly influence object detection, introduce an imbalance factor.

- Although the prediction's uncertainty is a suitable fit for an informativeness heuristic, uncertainty-based methods are vulnerable to the problem of data bias, since the picked data rarely accurately represent the entire unlabeled dataset. Furthermore, by introducing the uncertainty of box prediction, the acquisition function's scores may be altered depending on how many objects are in each sample.
- Given the indication that deep object detection algorithms yield superior performance with batch-based sampling techniques compared to conventional one-by-one sampling techniques, there are still other training-related challenges that need to be resolved:
  - In contrast to classification annotation, the annotation for object detection tasks is a more costly process that involves labeling each object in an image with a bounding box and its respective category.
  - Picking images that can significantly boost model performance is difficult, especially when trying to avoid outliers and noisy instances that could compromise the final results, as is the case with classification uncertainty.

To overcome the cost-accuracy trade-off, active learning algorithms designed for object detection can therefore still be expanded. These improvements depend on the choice of artificial network architecture and involve selecting the most appropriate sampling strategy [183].

3.2 Deep Active Learning

To effectively optimize the extensive parameters inherent in DL models, there's a significant need for large datasets that enable the extraction of high-quality features. Nonetheless, the challenge of obtaining enough training data persists, complicated by issues like class imbalance, the high costs of annotation, and the dynamic nature of datasets, making it difficult to collect a substantial amount of data for training deep models efficiently. To overcome this challenge, AL appears as a natural solution.

However, conventional AL methods were originally designed for various machine learning models. Yet when AL is applied to deep learning, it faces significant difficulties. This is especially true for complex structures and high-dimensional input data while complicating the training process. With the emergence of deep AL (DAL), a combination of traditional AL with DL, numerous novel approaches to implementing AL have emerged to co-exist with current applications. This section provides an overview of this context.

3.2.1 Generic Deep Active Learning Framework

A key feature of Deep AL is its strong emphasis on batch-based sample querying, which mitigates the problem of excessive model updating that arises with per-sample query methods, and effectively balances the costs of sampling and training. Starting from already-existing labeled data, the deep model is updated initially in an active learning cycle. Subsequently, by evaluating the remaining unlabelled data, features extracted, or the outcomes of the obtained deep model, the query strategy is used to actively select the most informative batch from the unlabeled pool. In this way, it is anticipated that the deep model's performance will converge through multiple iterations of interaction between active learning and model training. Each cycle offers the choice to update the deep model or conduct fine-tuning using newly collected data exclusively.

Based on the pillars of Exploitation and Exploration, a general deep active learning framework is occasionally outlined as follows [185].

Over T cycles, the two steps are iteratively interchanged to enhance a deep learning model.

1. **Exploitation step (model training)**: Given a labeled sample $\mathcal{U}_{ann} = \{(x_i, y_i)\}_{i=1}^{M}$ and a deep model $f(\cdot;\theta_t)$ with parameters θ_t, to be optimized in the t-th cycle, the exploitation step consist of training the baseline $f(\cdot;\theta_t)$ using $\mathcal{U}_{ann}$ to minimize the loss function L:

$$\theta_t = \arg\min_{\theta} \sum_{\{x,y\}\in U_{ann}^{t}} L(f(x;\theta), y) \tag{3.6}$$

2. **Exploration step (sample selection)**: In the exploration step, each sample in the unlabeled pool $U_{unann} = \{(x_j)\}_{j=1}^{N}$ is evaluated for its informativeness:

$$B_t = \arg\max_{B\subset U_{unann}^{t}} \sum_{x\in B} \text{Score}(x, \theta_t) \tag{3.7}$$

After evaluation, a batch of top-ranked samples B_t is chosen for requesting labels from the human expert (oracle). The labeled set is then updated by adding the labeled B_t, for retraining the model in the subsequent DAL cycle:

$$U_{ann}^{t+1} = U_{ann}^{t} \cup B_t \tag{3.8}$$

Generally, DAL methods aim to address two main challenges:

1. Selecting the most valuable batch of samples for labeling.

2. Training a deep model effectively with limited labeled data.

Additionally, these methods have been optimized in various perspectives as explained in the following subsection.

3.2.2 Enhancing of DAL Methods

- Automatically designed query strategy: The literature points out that the success of active learning is primarily determined by the deep model used, the nature of the data distribution, and the suitability of the active selection strategy to these factors [178]. Within deep learning, hand-built acquisition functions have shown several drawbacks. These approaches often adopt methods from conventional AL, designed for ML shallow models, and thus may not align well with the complexities of deep learning models. Moreover, their reliance on human expertise can limit their effectiveness,

To tackle the limitations of manual querying strategies, an intuitive solution is to create strategies that can autonomously learn and adapt their acquisition functions through interaction with unlabeled data. This shift towards automated learning strategies is encapsulated in the idea of "learning to learn actively." Such strategies leverage meta-learning to develop mechanisms that can decide on the most informative data points to label. These selection strategies fall under several categories of meta-learning techniques, including optimization- and metric-based methods.

Additionally, integrating the active learning challenge within a reinforcement learning framework has emerged as a novel strategy. Here, the acquisition function is treated as a policy that reinforcement learning can optimize. This combination of reinforcement learning with Deep Active Learning enables the querying strategies to adjust dynamically, offering a solution when the effectiveness of existing knowledge is uncertain. In this context, both value- and policy-based methods are prevalent for implementing reinforcement learning in Deep Active Learning scenarios [178, 186].

- Model training: Most DAL algorithms have been recently tailored for supervised learning scenarios, neglecting the rich information embedded within the vast amounts of available unlabeled data. Transfer learning, unsupervised learning, and semi-supervised learning are the most likely learning schemes used today to fully exploit the utility of abundant unlabeled data in a label-efficient manner. It is therefore logical to explore DAL in conjunction with these approaches to enhance model performance. While DAL emphasizes finding meaningful subsets of data with constrained annotation resources, integrating transfer learning with DAL is an intuitive approach to align well and boost annotation efficiency, demonstrated through applications like active fine-tuning, ADA, and active distillation, showcasing the effectiveness of this synergy [186].

- Deep Bayesian active learning: Generally speaking, representing the uncertainty is crucial in any active learning framework, however, deep learning methods are not capable of either representing or manipulating the model uncertainty. On the other hand, from the real-world application perspective, uncertainty representation is getting more and more attention in the machine learning community. Deep Bayesian active learning frameworks and generally any Bayesian active learning settings, provide practical consideration in the model which allows training with small data while representing the model uncertainty for further efficient training [187].

3.3 Deep Active Learning for Object Detection in Autonomous Driving: State-of-The-Art

3.3.1 Active Learning for Deep Object Detection Architectures

Regarding the deep investigation of DAL approaches for solving object detection problems, the existing works can be systematically classified based on various criteria as summarised in Figure 3.4.

As shown in Figure 3.4, key differences between these approaches reside in different aspects, including learning tasks, data and embeddings distribution, deep model predictions, query strategy design, selection criteria quantification (uncertainty, diversity, inconsistency, and label correlation), scoring level (pixel, box, region or image) and metric measurement, aggregation techniques, sampling granularity, labeling source, and data accessibility.

Unlike the typical AL methods, which target shallow models like SVM, advanced DAL-related researches are Widely explored DNN models, where CNNs are well-studied deep learning models. In most heuristic-based DAL approaches, Uncertainty Sampling (US) [10], [188, 189, 190, 191], Query-by-Committee (QBC) [192], mutual information [193], and expected model change [194] are most heuristics used, as a single criterion, throughout the query strategy to select a single instance at a time. These heuristics can be assessed and measured using either the training model itself or a separate model. However, several researchers have reported the limitations of applying a single-criterion, per-instance query strategy in supporting the batch training approach inherent in DL methods. Furthermore, in the supervised training scenario of a deep model, only labeled data is accessible during the DAL cycles, without any assistance from the remaining unlabeled data.

To deal with these limitations, multiple selection criteria are considered for enabling efficient CNN-based model training across AL cycles. Therefore, hybrid-criteria, mixture-criteria, multi-criteria [195] or batch-based sampling [196, 197] query were proposed to select a substantial amount of samples to be labeled at a time while attempting to find a balance between the considered strategies.

Moreover, promising research directions have been explored to extend DAL algorithms

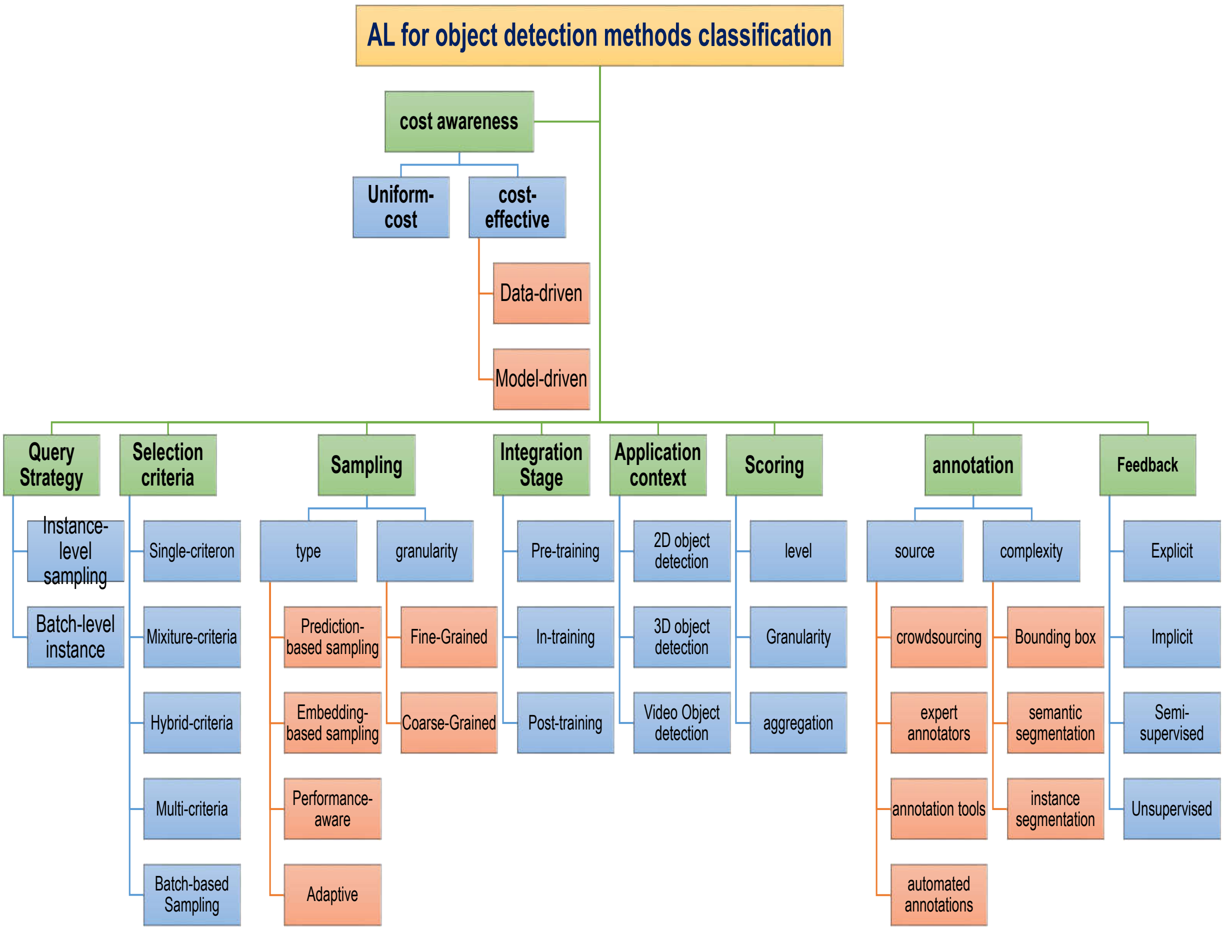


Figure 3.4: A taxonomy of active learning for deep object detection architectures.

regarding the integration of different annotation granularity, abundant unlabeled data, and related supervision setting into active learning pipeline, including multi-label [198], multi-view [199], multi-instance [189], multi-instance multi-label (M2AL)[200], multi-view multi-instance multi-label (M3AL) [201], and unsupervised [202], [193] AL schemes. Among them, more attention has been paid to address two aspects: the automatic design of selection samples strategy [203] and the alleviation of various problems, namely data-related problems such as confidence and insufficient labeled sample, model-related problems such as generalization ability, and domain-specific problems such as domain shift, cold-start problem, and class imbalance.

Besides, serious research has been conducted in recent years to properly design cost-effective DAL frameworks. The main idea is to adopt both query-driven and data-driven cost-saving strategies. The query-driven approaches were based on gaining support from complementary techniques to perform query improvement, such as optimization techniques, metrics learning [204], and alternative learning paradigms (one-shot, contrastive, federated, goal-driven, domain adaptive...) [205, 206, 207, 208, 209, 210]. On the other side, data-driven approaches were attempted to address several data-level perspectives in terms of data labeling supervision (weak, self, semi...) [190, 211, 212, 213], labeling setting (open-set recognition) [214, 215] and granularity [197, 200]. For further details please refer to the survey papers [216, 180, 217].

The next subsections describe related work on the latest DAL approaches that employ CNNs for object detection tasks in general and autonomous driving applications in particular.

3.3.2 Active Learning for Deep Object Detection

An uncertainty-based active learning approach for object detection in remote sensing images is presented in [188]. The authors argue that an efficient weighted combination of classification and regression uncertainty could overcome class imbalance and object density variation difficulties. Based on predictions (bounding box and classification probability) of a CNN-based detector on unseen, unlabeled images, the high-ranked image could be selected according to the image-level uncertainty score aggregated by summing each object uncertainty within the unlabeled image. With the low granularity level, the authors in [189] explored the instance-level for object detection. Throughout a multiple-instance unsupervised active learning approach, the unlabeled images are treated as instance bags and feature anchors in images as instances where the image uncertainty is estimated using instance uncertainty learning and instance uncertainty re-weighting modules. As a result, the high-ranked images are used to train a constructed detector based on RetinaNet. By adopting query by committee, Roy et al. [192] formed a committee of classifiers by leveraging extra detection head layers of the deep network architecture (SSD). As a selection criterion, the disagreement is measured and aggregated by introducing the 'margin' for each bounding box. By considering mAP improvement and class imbalance between background and object categories, li et al. [196] proposed WBetGS that enhances typical diversity and uncertainty-based batch sampling for batch mode active learning in object detection. Nevertheless, inefficient training of CNN-based detectors, redundant data selection, scalability handling, and heavy burden of convergence time are the main shortcomings.

A review of existing research on cost-effective DAL for object detection is relatively sparse. Most of these works are built upon mixed supervised learning methods. Leveraging access to both labeled and unlabeled data, a supervised signal is provided which optimizes iterative DAL cycles and reduces human annotator [190, 211, 212, 213]. Wang et al. [211] proposed an active sample mining (ASM) framework for cost-effective training of object detectors. Focusing on a switchable sample selection mechanism, several unlabeled samples are selected, according to deep detector predictions, to automatically pseudo-label via a novel self-learning process. However, the remaining samples are manually annotated via an active learning process. For cost-effective panicle detection in cereal crops, the authors in [190] proposed an uncertainty-based active learning approach suitable for two-stage models. Only strong labels (tight bounding boxes) are queried by considering high uncertainty images picked from a constructed low-cost weak labeled (object center clicking) subset driven by the oracle labeling knowledge. Alternatively, some works focus on exploring other metrics, such as consistency and entropy, to evaluate model predictions between the original and augmented data [218, 219].

3.3.3 Active Learning for Deep on-road Object Detection Architectures in Autonomous Driving

Ahead of deep active learning's breakthrough in vision-related operations for autonomous vehicles, numerous proposals of active learning methods, involving hand-crafted features and shallow classifiers, have targeted vehicle [220] and pedestrian [221] detection. Recently, few works have described deep active learning for on-road object detection.

Aghdam et al.[10] addressed pedestrian detection in images and video. Based on CNN-based object detector predictions, pixel-level scores are computed and aggregated as a single image-level score. Thus, a fixed number of high-ranked unlabeled images is selected for querying. With the introduction of temporal selection rules, the selection of highly visually similar video frames could be avoided.

Furthermore, the authors in [191] investigated LiDAR data and deep active learning for 3D object detection tasks. For training a LiDAR-based 3D object detector, they implemented an uncertainty-based approach that queries informative unlabeled samples from point cloud data, with the help of 2D region proposals in RGB images. Using the same data format, the authors in [222] explored localization-based uncertainty metrics for selecting samples from feature space, without relying on additional 2D input information. This proposed DAL method is built upon a specific object-matching process and is suitable for an anchor-based object detection architecture. Besides, Liang et al. [223] tailored the diversity metric by proposing a novel spatio-temporal diversity-based acquisition function that selects frames from a multimodal data pool. To ensure multi-view vehicle detection, the authors in [224] proposed an active learning algorithm to enhance the typical deformable part model by selecting B more effective part samples for query labeling by human annotator from multi-view vehicle images. Consequently, labeled part samples are considered as positive samples to retrain the SVM model as a learning part model.

To overcome the problem of training a model on decentralized data, the author in [210] explored various schemes to use uncertainty-based active learning with federated learning, where active learning is primarily used to label images locally, without transmitting the data to a central location, and then federated learning is used to train a global object detection model. Using the least confidence method for uncertainty sampling and the "sum" aggregation method, the results demonstrate that training the pre-trained YOLOv5 object detection model on the KITTI dataset achieves average precision levels close to centralized learning for homogeneous data.

- Comparison between different DAL for on-road object detection: This comparison seeks to investigate various DAL approaches aimed at improving the detection of on-road objects, both 2D and 3D, by examining several metrics and scorning techniques. Table 3.1 presents the reviewed works.

As seen in this table, existing works in active learning for on-road object detection improve the efficiency of training 2D/3D object detection models. Each work chooses specific metrics and data modalities to assess informative and representative samples by leveraging model architecture and output uncertainty.

The main problem of object detection in autonomous driving is noisy data (redundant, outliers) and annotation costs, which directly influence the model's generalization and training cost (i.e. overfitting). Achieving the desired performance requires determining the optimal samples for labeling based on computational efficiency, minimal redundancy, and savings in query and annotation costs.

Hence, many works have been proposed to decrease these drawbacks. Some works neglect the selection of of sample batches and label correlation, as seen in [210, 224, 222, 191], which are important factors for efficiently supporting deep model training. Other works neglect the variability of annotation cost, which also affects the selection/training processes by exceeding the annotation budget limitation.

By improving the inherently more efficient and scalable batch sampling strategy while optimizing batch selection based on cost awareness, we argue that leveraging the uncertainty of CNN-based detector predictions, the diversity of learned representations, and the adaptive selection strategy can help reduce the selection of redundant (noisy) samples, handle the variable cost, speed up the annotation process, and ultimately constitute an effective training set for building a competitive object detector while relaxing human supervision.

3.4 Challenges and Limitations

Despite advancements in deep active learning, there are still open issues that limit its use in practical vehicular environment scenarios [225].

- **Cold-start Problem**: Active learning operates in an iterative manner, choosing a fixed number of samples in every cycle and subsequently updating the model with the

Table 3.1: Comparison of DAL Frameworks for Object Detection in Autonomous Driving

Work	Key Features	Annotation Cost	Model	Model Performance	Dataset	Integration	Selection criteria	Uncertainty	Diversity	Scoring Level	Aggregation	Sampling Granularity	Target object	Application	data modality
Aghdam et al.[10]	-Focuses on selecting informative samples -labeling budget consideration -propose a new image-level scoring process -comparing softmax scores of adjacent pixels	fixed labeling budget	FPN-like network	Moderate (MR)	-CityPersons -Caltech Ped -BDD100K	Post-training (fine-tuning)	Single criterion	Classification Predictions	/	Pixel	average of max-pooled scores	Batch	Pedestrian	2D	images and videos
[191]	-Focuses on selecting informative samples, - Focuses on LiDAR 3D object detector, - Leverages 2D region proposals generated from the RGB images, - Propose a new image-level scoring process	Uniform cost	ConvNe-like network	High (Acc, MSE)	KITTI	Post-training (fine-tuning)	Single criterion	Classification Predictions (MC-dropout, Deep Ensemble)	/	Box	/	Instance	vehicle, Human	3D	LiDAR
[222]	- Focuses on anchor-based object detection architecture, - Focuses on LiDAR-based 3D object detector, - Modified object matching strategy, - Novel acquisition metric, -captures variability in numbers of objects detected for the same input	Uniform cost	VoxelNet-like network	Moderate (mAP)	Government owned	Post-training (scratch)	Mixture criteria	Classification and regression Predictions	/	Box	averaged the confidence scores	Instance	various	3D	LiDAR
Liang et al. [223]	- Focuses on selecting informative and diverse samples, - Take advantage of the multimodal information, - Novel spatio-temporal diversity-based acquisition function, - Considering costs for annotating both a frame and a 3D bounding box - Cold-start problem	Cost effective	VoxelNet network	high (mAP)	nuScenes	Post-training (fine-tuning)	Mixture criteria	/	distance measure	Box	/	Batch	various	3D	multimodal
[224]	- Focuses on selecting informative samples, - Focuses on Multi-view Detection based on Part Model, - Collect part samples	Uniform cost	/	High (Acc)	CBCL street scenes	/	Single criterion	/	/	Pixel	/	Instance	vehicle	2D	images
[210]	-Focuses on selecting informative samples, -Focuses on federated learning and Active learning for training on decentralized data, - A chain of devices allows for increased precision	Uniform cost	YOLOv5 object detection	Moderate (mAP)	KITTI	Post-training (fine-tuning)	Single criterion	Classification Predictions (CS)	/	Box	sum of Confidence	Instance	vehicle, Human	3D	LiDAR

annotated samples accumulated so far. Within this process, two primary challenges arise: (1) determining the initial set of samples to train the first model and start the iterative process, and (2) deep learning models' tendency to produce unreliable probability estimates when trained on sparse data. These challenges, known as the cold-start problem, highlight the necessity for labeling a sufficiently large initial subset to start the AL cycle. This problem is particularly pronounced for deep learning models due to their substantial data requirements.

- **Combining informativeness and representativeness**: Classical acquisition functions typically focus on optimizing informativeness or representativeness objectives, each serving a distinct purpose in the sample selection process. Combining the two objectives can be challenging due to the differing nature of the selection strategies. The informativeness measure selects samples that are likely to add missing information to the model, based on the predicted probability distribution. The representativeness measure selects samples to well represent the overall input patterns of unlabeled data, based on their representation in the embedding space. The two measures convey complementary information and a few sets of approaches have attempted to tackle their combination to select samples that are both representative and informative. While such combinations could help in selecting more valuable samples, joint optimization remains an open issue.

- **Imbalanced learning in AL**: Generally speaking, the class imbalance problem is present in most practical datasets.Although active learning has the potential to address class imbalance, it has been shown that, unless properly processed, a high degree of imbalance can adversely affect the selection process, with the biased model having a preference for selecting samples from majority classes. Active learning for an imbalanced dataset adds another component to the selection criteria to ensure that the imbalance in the unlabeled set is not transferred to the selected set.

- In many crowd-sourcing settings, users may be unreliable, makeing accurate assessments of annotation quality challenging. Furthermore, once obtained, these assessments may not be reusable, as the model lacks control over their selection. Research on using non-experts or even unreliable experts as oracles in active learning remains in progress.

- The difficulty with batch-mode active learning lies not only in ensuring scalability to large datasets and high-dimensional data but also in balancing exploration and exploitation while minimizing computational and complexity costs. This involves correctly assembling the best query batch Q. Additionally, while it may reduce the number of instances needing annotation, it may not necessarily decrease the overall annotation time.

- **Noisy Labeled Data**: Noisy, incorrectly labeled data may alter training data and carry out greater harm than just having insufficient training data. Expert error or inadequate knowledge to label new data due to restricted information are some of the reasons for noisy points. Open research questions regarding noisy labeled data include how to handle situations where no experts know the ground truth and how active learners can deal with experts whose quality varies over time [178].

- **Low Query Budget and Variable Labeling Costs**: An important aspect of active learning is that significant amounts of unlabelled data should be queried before satisfactory results can be obtained. However, labeling costs and time often result in a limited budget for querying. In practice, large starting labeled datasets and a costly query are usually needed for deep active learners to efficiently optimize a massive number of model parameters, particularly when dealing with high-dimensional data. However, it becomes difficult to obtain enough data for accurate deep model training when the query budget is constrained. Furthermore, in autonomous driving applications, both labeling quality and cost can vary significantly between samples. This means that reducing the querying budget may not always affect the total labeling cost. Therefore, the problem here is how to accurately assess labeling costs under these circumstances [178].

- **AL with Outliers**: Data points that significantly differ from the dataset's average value are referred to as outliers. Querying these outliers can result in inefficient use of labeling resources through examination of areas that diverge significantly from typical data, thus negatively impacting the detection model's performance. Therefore, the problem here is how to handle the presence of outliers [178].

- **Combination inconsistency**: In general, AL aims to train the detector model by using fixed feature representations. Conversely, feature learning and detector training are optimized together in the deep learning process. As a result, integrating DL and AL is very challenging due to the inconsistency in their processing pipelines, while other research either addressed them as a pair of issues or limited their attention to updating the deep detector model throughout the AL framework.

- **Domain shift**: Domain shift refers to the inconsistency between the distribution of labeled training data and unlabelled data. In the context of conventional Deep AL, labeled and unlabelled data are typically assumed to have the same distribution. This leads to domain shift due to performance degradation when the target data originate from a different domain in the actual world [186].

Conclusion

This chapter has thoroughly explored various deep active learning methods for object detection in autonomous driving. We discussed the fundamental concepts, classification, and comparison of these methods based on different criteria to ensure high-performance and robust object detection. Our focus on batch-mode deep active learning demonstrated the potential for reducing labeling costs while maintaining high accuracy. The review highlighted the importance of efficient query strategies and the challenges posed by noisy oracles and domain shifts. These insights set the stage for the next step in our research—developing a cost-effective framework for training deep on-road object detectors.

In the following chapter, we will introduce our novel contributions to this field, aiming to enhance the efficiency and effectiveness of deep active learning in practical applications.

Chapter 4

Methodology

Introduction

Toward reducing the training cost of a deep object detector with deep active learning, literature review findings in the previews chapter indicated that a batch-based query strategy within cost-effective DAL is an efficient approach to alleviate labeling and training cost variants under given budget and desired performance constraints. This label-efficient iterative learning algorithm is suitable for fine-tuning a deep object detection model, considering a labeled batch as a min-batch.

Building upon concepts and methods discussed in the previous chapter, this chapter presents the design and implementation of our cost-effective deep batch mode active learning framework (CEDBMAL). Section 4.1 details the proposed uniform-cost batch-based query strategy to improve deep batch mode active learning for object detection, describing proposed active learning metrics and scores based on classification and regression prediction uncertainty, object distribution, and diversity, allowing active selection of fixed-size batches queried across sequences of frames. Subsequently, it presents the cost-effective version of this strategy to handle variable labeling costs and batch size setting issues, reviewing the basis used in this contribution, including labeling time prediction as an annotation cost estimator and the 0-1 Knapsack algorithm as optimal batch size selection solver. As a result, the proposed training framework integrates the enhanced query strategy with the transfer learning scheme to build an efficient and robust single-stage CNN-based object detector. Section 4.2 provides extensive experiments to validate our two contributions, introducing the target dataset, the underlying CNN-based object detector, the training scheme, and settings. It also illustrates and discusses both quantitative and qualitative results of performance evaluation, as well as comparisons with similar solutions.

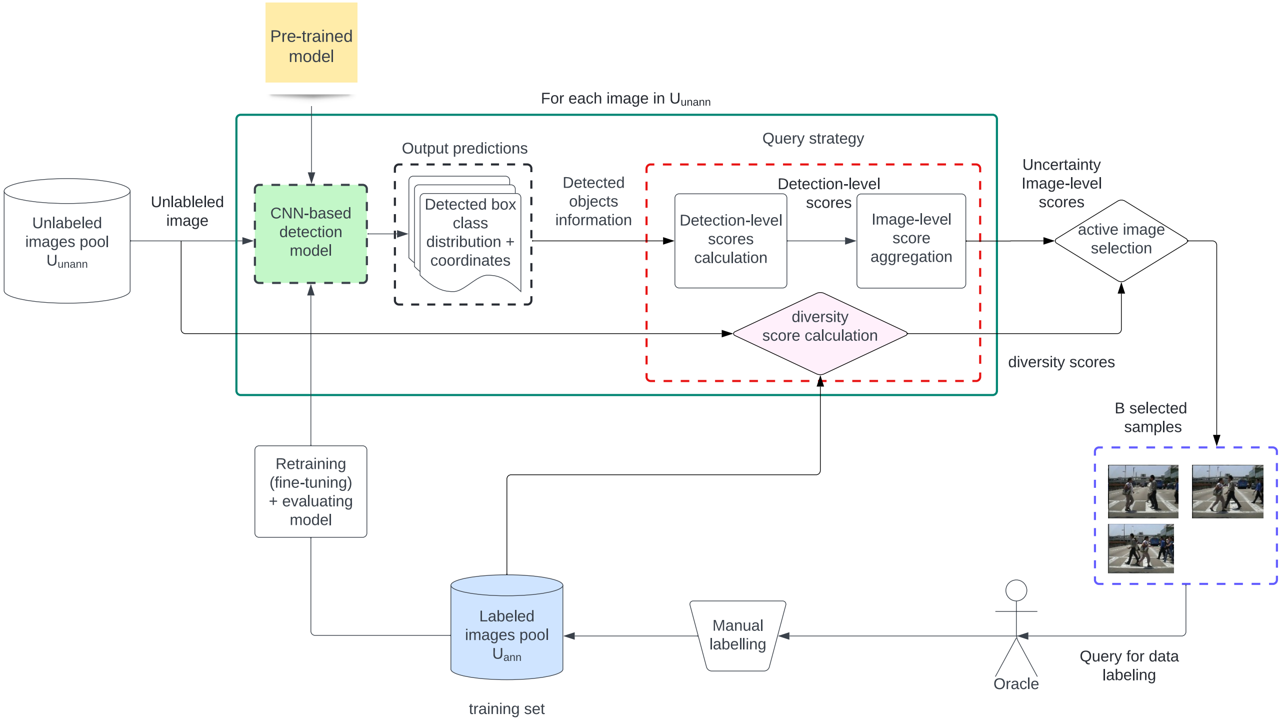

Pre-trained model
For each image in U_{unann}
Output predictions
Query strategy
Unlabeled images pool U_{unann}
Unlableled image
CNN-based detection model
Detected box class distribution + coordinates
Detected objects information
Detection-level scores
Detection-level scores calculation
Image-level score aggregation
Uncertainty Image-level scores
active image selection
diversity score calculation
diversity scores
B selected samples
Retraining (fine-tuning) + evaluating model
Labeled images pool U_{ann}
training set
Manual labelling
Oracle
Query for data labeling

Figure 4.1: Overview of the proposed deep active learning framework

4.1 Overall Framework

CEDBMAL framework focuses on a pool-based setting that consists of an iterative selection/annotation process as depicted in Figure 4.1. Given a large pool of unlabeled images U_{unann} and a labeling budget, CEDBMAL initially employs the underlying pre-trained detection model to examine each unlabeled image and then selects, using a query strategy, batches of more valuable examples based on uncertainty as an informativeness measure and diversity as a representativeness measure. Next, a batch with the best size value, among the selected batches in the first step, is picked out for manual labeling by leveraging information from the resolved 0-1 Knapsack problem, considering low redundant instances with more objects of interest and fewer estimated annotation time. Once labeled by an oracle (e.g., human annotator), the labeled images pool U_{ann} is enlarged with this labeled subset, which is retired from U_{unann}. These accumulated actively-labeled images are considered a training set to fine-tune the detector while getting an updated model in the result. The cycle of these steps continues on the remaining unlabeled images until the labeling budget is exhausted or the required performance is achieved cost-effectively. The next subsections describe the detection model followed by our framework's cost-effective active learning strategy.

4.1.1 Detection Model

This work focuses on a single-stage CNN-based object detector as a state-of-the-art object detector. Such a model relies on a baseline CNN model for feature learning and extra head layers for object classification and bounding boxes regression. The overall deep architecture is trained end-to-end, with a post-processing method obtaining the final detection outputs. For detector prediction, the 2D map of probabilities per class and bounding box coordinates are used to rank examples in recent uncertainty-based deep active learning works [226]. The pre-trained object detector is fine-tuned using a transfer learning paradigm instead of training from scratch to reduce training costs and explore the domain-shift influence on overall performance.

4.1.2 Deep Active Learning for CNN-based Object Detector

To train the underlying detection model, the active learning method should carefully employ a properly designed query strategy for querying labels while identifying the cost of the selection/annotation process. Independently of the underlying detection model architecture, our query strategies are performed as explained below.

- **Uncertainty-based deep active learning**

Despite the effectiveness of uncertainty-based DAL for classification tasks, it needs to be revised for object detection. These selection strategies suffer from querying outliers. They are less efficient in evaluating image data in autonomous driving datasets when solely relying on the predicted class uncertainty of the CNN-based model. Therefore, selecting more valuable unlabeled images may fail, negatively impacting detection performance. To address these challenges, we suggest incorporating regression alongside classification in an uncertainty sampling strategy, as explained below.

- Classification uncertainty sampling: Given an example x, a CNN-based object detector estimates the probability distribution of the label $p(c|x)$ over C classes per detected bounding boxes. Such predictions are evaluated by a scoring function to measure the uncertainty metric and form detection-level scores for each detected object, using uncertainty sampling for this purpose. For a given bounding box Bb, its classification uncertainty $U_C(Bb)$ is defined as $U_C(Bb) = 1 - P_{max}(Bb)$ where $P_{max}(Bb)$ is the highest probability distribution among all classes.

- Regression uncertainty sampling: Since CNN-based detector predicts bounding box coordinates, regression uncertainty can be measured by estimating distribution probability density [188]. Adopting the Gaussian Mixture Model(GMM), each bounding box's distribution probability density (denoted L) is estimated in terms of calculated log probability. The obtained L is clipped as Lb=min(-99, L). Finally, the regression uncertainty (U_r) is calculated using the following uncertainty formulation:

$$U_r = \begin{cases} 0.05 * (Lb + 10) + 0.5, Lb \geq -10 \\ 0.5 * \frac{Lb+100}{90}, Lb < -10 \end{cases} \tag{4.1}$$

- WCR Deep Active Learning: Inspired by [188], our proposed weighted classification-regression (WCR) uncertainty-based deep active learning algorithm uses both classification uncertainty U_c and regression uncertainty U_r to perform the query strategy. However, an unlabeled image is not selected for querying unless the WCR image-level uncertainty, denoted as U_s, is aggregated from detection-level scores for each detected box (object) in it, as:

$$U_s = agg(U_c(Bb) \times U_r(Bb))$$

, where Bb $\subseteq$ detected Bboxes. In our work, the aggregating methods are performed as inspired by [226].

- **Sum**: Given an unlabeled image x, the aggregate score, from the detected bounding boxes D, can be obtained as follows

$$v_{Sum}(x) = \sum_{Bb \subseteq D} U_C(Bb) \tag{4.2}$$

- **Average**: With less sensitivity to the number of detections, the main idea is averaging all detection-level scores.

$$v_{Avg}(x) = \frac{1}{|D|} \sum_{Bb \subseteq D} U_C(Bb) \tag{4.3}$$

- **Maximum**: The maximum of detection-level scores is kept. Despite the robustness of zero-valued detections (as noise), a substantial information can be lost.

$$v_{Max}(x) = \max_{Bb \subseteq D} U_C(Bb) \tag{4.4}$$

According to the one-by-one query method, the query function can select a group of B unlabeled images with higher WCR uncertainty while ignoring outliers. Our contribution is highlighted in algorithm 4.1.

Despite its robustness, this solution could select redundant images, less effective in the training process. This limits capturing the visual pattern diversity in typical urban road scenarios. Additionally, the repetitive one-instance-at-a-time selection procedure can lead to an inefficient and time-consuming training process, placing an expensive burden on the annotator expert. Other issues include impractical settings of B (ranging from increasing

Algorithm 4.1 WCR deep Active Learning Algorithm Implementation Details

```
Require: annotated images pool $U_{ann}$, unannotated images pool $U_{unann}$, object detector $OD$, test set
    $U_{test}$, objects' categories $C$
$U_{ann} \leftarrow \emptyset$
$OD \leftarrow$ pre-trained object detector $OD_0$
repeat
    for each image $x$ in $U_{unann}$ do
        Fed $x$ into the object detector $OD$
        Get bounding boxes $Dx$ with corresponding posterior probability p(c|Bb) and coordinates after
    post-processing operation (NMS)
        for each object $Bb$ in $Dx$ do
            Use objects' information to calculate $U_c$ and $U_r$
        end for
        Calculate WCR uncertainty $U_s$ using each $x$ object's $U_c$ and $U_r$
    end for
    Sort $U_{unann}$ (in descending order) using the assigned WCR uncertainty $U_s$ scores
    Select B high-ranking images as queries for annotation by an oracle
    $U_{ann} \leftarrow U_{ann} + B$, $U_{unann} \leftarrow U_{unann}\breve{}B$
    $OD_t \leftarrow OD_{t-1}$ fine-tuned on $U_{ann}$
    Test $OD_t$ using $U_{test}$
    Evaluate the detection performance (detection loss)
until The required performance is reached or query budge
Ensure: detector model parameters $W_F$ and the final detector model $OD_F$
```

time-to-completion to uniform sampling of images) and fixed assumptions about labeling cost. To address these issues, incorporating our proposed uncertainty measure in a batch sample query strategy can ensure cost-effective training and labeling tasks, where true diversity within a group of instances is guaranteed.

- **Cost-effective deep batch-mode active learning**

Two critical design points, namely batch query and batch size selection, are carried out as explained below.

- Uniform-Cost Deep Batch-mode Active Learning: Compared to one-by-one query strategies, several deep batch mode active learning researches have shown the efficiency of hybrid batch-based query strategies in training CNN-based object detectors [216]. In this setting, the final score used for ranking the unlabeled images and picking diverse samples with high uncertainty is calculated as [227]

$$finalScore = \alpha \times (1.0 - similarityScore) + (1.0 - \alpha) \times uncertaintyScore$$

, where the parameter α weights the impact of each factor as:

$$\alpha = \frac{|U_{unann}|}{|U_{unann}| + |U_{ann}|}$$

In this work, we investigate the previously presented WCR uncertainty as an informativeness criterion, while choosing Euclidean distance as a similarity measurement.

This method favors the selection of the furthest unlabeled sample x_i from its closest labeled neighbor, where the distance between them is computed as follows [228]:

$$div_i = \min_{j=1,2,,,,n} ||x_i - x_j||^2, x_i \subseteq U_{unann}, x_j \subseteq U_{ann}$$

However, one of the fundamental issues in using a batch-based query strategy lies in the batch size, which might produce worse results and make the labeling effort inefficient [227]. To relax this limitation, our contribution consists of selecting a diverse batch with optimal size at each iteration under given budget and desired performance constraints.

- Cost-effective Deep Batch-mode Active Learning: In autonomous driving context, the selection of an optimal batch of instances that positively impacts detection performance is driven by determining the batch size, which ensures an adaptive response to varying labeling time. Regardless of a particular batch size and inspired by [229], the optimal batch size selection is reduced to the 0-1 Knapsack problem [230], which maximizes the uncertainty, maintains the annotation costs within a given budget and desired performance constraints and, can be solved with dynamic programming. First, we pick a set of batches with size Q_i from unlabeled images pool, where $Q_i \subseteq 100 \ldots$ | unlabeled images pool |. Given such a batch set, where each item has a weight T_i and value V_i, a 0-1 Knapsack problem is formulated. It's worth noting that the batch uncertainty V_i is defined by summing the uncertainty of top-Q_i images within the batch i.e.

$$V_i = \sum_{j=1}^{Q_i} V_{ij}$$

while the labeling time is predicted as annotation cost T_i. Figure 4.2 describes this combination.

Algorithm 4.2 depicts the overall operations of the 0-1 Knapsack algorithm using dynamic programming.

Labeling time prediction: As cited in [190], the annotation time for baseline methods, given a batch of queried images, is calculated using the following formula

$$T_i = 7.8 \times Q_i + 34.5 \times bQ_i$$

where Q_i is the batch size, bQ_i is the total objects in it and $T_i <$ T. In our work, bQ_i is the total number of the predicted BBox within the batch.

As a result, the most useful instances with low redundant information can be selected for labeling, improving the performance at every iteration and saving immense labeling loads given a fixed budget. The overall operations in our CEDBMAL are depicted in algorithm 4.3.

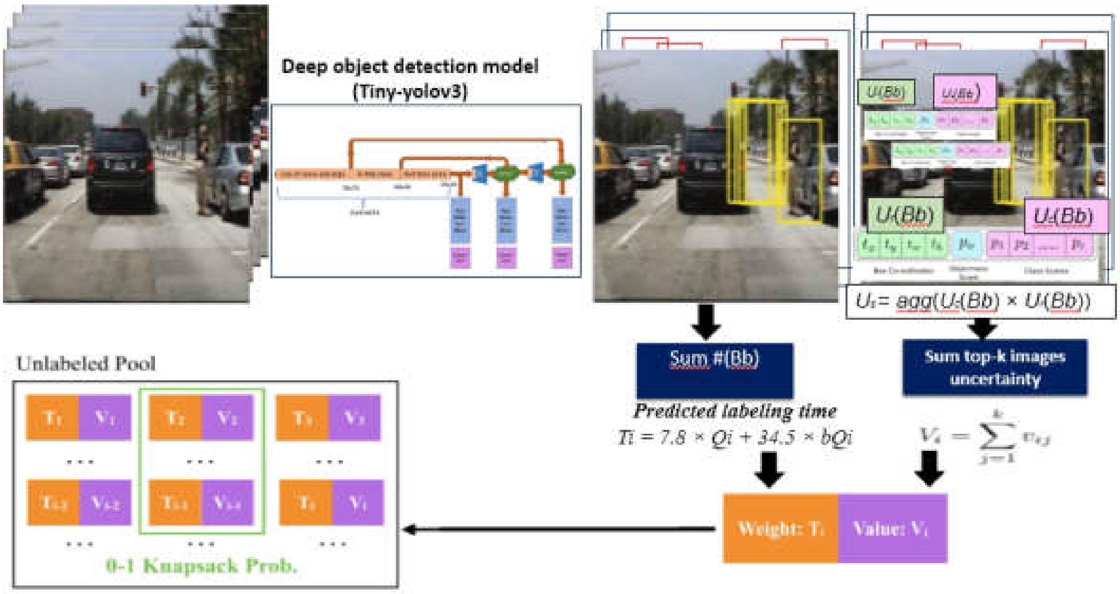


Figure 4.2: 0-1 Knapsack problem formulation with T_i as Weight parameter and V_i as Value parameter.

Algorithm 4.2 0-1 Knapsack Algorithm using Dynamic Programming.

```
Require: Array of weights $w$, array of values $v$, number of items $n$, maximum capacity $W$
Function KNAPSACK($W, n, w, v$)
Create a 2D array $K[n+1][W+1]$
for $i = 0 \to n$ do
  for $w = 0 \to W$ do
    if $i == 0$ or $w == 0$ then
      $K[i][w] \leftarrow 0$
    else
      if $w_i \leq w$ then
        $K[i][w] \leftarrow \max(v[i-1] + K[i-1][w - w[i-1]], K[i-1][w])$
      else
        $K[i][w] \leftarrow K[i-1][w]$
      end if
    end if
  end for
end for
Return $K[n][W]$
End Function
Ensure: Maximum value that can be obtained within capacity $W$
```

Algorithm 4.3 CEDBM Active Learning Algorithm Implementation Details

```
Require: annotated images pool $U_{ann}$, unannotated images pool $U_{unann}$, object detector $OD$, test set $U_{test}$,
    objects' categories $C$
$U_{ann} \leftarrow \emptyset$
$OD \leftarrow$ pre-trained object detector $OD_0$
repeat
  for each batch size $Q_i$ in 100 ... $|U_{unann}|$ do
    for each image $x$ in $U_{unann}$ do
      Fed $x$ into the object detector $OD$
      Get bounding boxes $Dx$ with corresponding posterior probability p(c|Bb) and coordinates
      for each object $Bb$ in $Dx$ do
        Use objects' information to calculate $U_c$ and $U_r$
      end for
      Calculate WCR uncertainty $U_s$, as $UncertaintyScore_x$, using each $x$ object's $U_c$ and $U_r$
      Calculate $similarityScore_x$ using Euclidean distance
      calculate $score_x = \alpha\times$ (1.0 - $similarityScore_x$) + (1.0 - $\alpha$) $\times UncertaintyScore_x$
    end for
    Sort $U_{unann}$ (in descending order) using the assigned $scores_x$
    Select a batch of instances $B_{Qi}$ with largest $score_x$
    $bQ_i \leftarrow 0$
    $V_i \leftarrow 0$
    for each image $x$ in $B_{Qi}$ do
      $bQ_i \leftarrow bQ_i + D_x$
      $V_i \leftarrow V_i + UncertaintyScore_x$
    end for
    Estimate the annotation time $T_i$ (as cost) using $Q_i$ and $bQ_i$
  end for
  Estimate the optimal batch size by solving a 0-1 Knapsack problem using $T_i$ and Uncertainty $V_i$ for each
    batch size
  Select the batch $B_{best}$, with the best batch size, as queries for annotation by an oracle
  $U_{ann} \leftarrow U_{ann} + B_{best}$, $U_{unann} \leftarrow U_{unann}\breve{}B_{best}$
  Fine-tune $OD_{t-1}$ using $U_{ann}$ to get $OD_t$
  Test $OD_t$ using $U_{test}$
  Evaluate the detection performance (detection loss)
until The required performance is reached or query budge
Ensure: detector model parameters $W_F$ and the final detector model $OD_F$
```

4.2 Experiments and Results

4.2.1 Experimental Setup

To study how our DAL framework could ensure a cost-effective annotation and training processes while reducing manual annotation effort and guaranteeing expected detection performance over an autonomous driving dataset, we use it to fine-tune a pre-trained Tiny-YOLOv3 for detecting pedestrians (as a use case) on the Caltech Pedestrian Detection Benchmark [156] while evaluating various setting of B. In our experiments, we retain only training frames labeled as "person" with a height taller or equal to 20 pixels, simulating the oracle annotation and approaching safety risk assessment of the trained model using partial specifications [22]. For evaluating the detector's performance (detection loss), we use the test set and Piotr's Matlab Toolbox, providing a fair and comprehensive comparison against two other alternatives: transfer learning and random sampling. In this case, the "Reasonable" scenario is preferred. For validation purposes, we split the training set by 10% as a validation set. The target dataset, the tiny version of the detection model, and the alternative sampling methods are discussed in detail below.

- **Dataset**

The Caltech Pedestrian dataset [156] consists of $\sim$ 10 hours of 640x480 30Hz urban driving video with 350K labeled bounding boxes whereas 2,300 unique pedestrians were annotated. Over the 11 sessions, it resulted in 42,782 training images (set00-set05) and 4,024 test images (set06-set10) sampled every 30th video frame. The log-average miss rate is used to evaluate the detection performance and is calculated by averaging the miss rate on false positive per-image (FPPI) points where the relevant point is defined as FPPI $= 10^{-1}$. 4 testing scenarios which are "All", "Reasonable", "Scale=near", and "Scale=medium" are defined. The statistics of the frames with bounding boxes labeled as "person" with a height of 20 pixels are summarized in TABLE 4.1.

Table 4.1: "person" labeled frames statistics in train and test sets

# unlabeled frames		**"Person" label**	
Train	**Test**	**# labeled images**	**# Bounding Boxes**
4250	4024	2006	4987

- **Tiny-YOLOv3**

In this work, we focus on Tiny-YOLOv3, a simplified version of YOLOv3 [121]. Emphasizing the Darknet-53 backbone and its low-complexity architecture motivates its suitability for constrained environments with a significant detection speed but at the cost of some detection accuracy loss. In our experiments, we use a pre-trained version of the Tiny-YOLOv3 on the COCO benchmark [231], containing 82 object categories.

We also perform two training scenarios. Firstly, we freeze the Darknet backbone layers and fine-tune the other layers while setting training parameters as follows: learning rate: $1e^{-3}$, number of iterations: 60, mini-batch size: 16. Next, we unfreeze all layers and fine-tune them using the following training parameters: initial learning rate: $1e^{-4}$, initial number of epochs: 60, number of epochs: 120, batch size: 16. For both scenarios, we use Adam as the optimization algorithm while the learning rate decays by a factor of 0.1.

- **Random Sampling**

Random sampling, as passive learning, is a naive sampling technique that aims to choose the frame to be labeled uniformly at random from an unlabelled pool. The selected frames are therefore independent and not known beforehand [232].

- **Transfer Learning**

Generally applied in deep learning, transfer learning focuses on the transfer of knowledge from source domains to target domains while fine-tuning a pre-trained deep model. Thus, the performance of the deep model could be improved by exploiting parameter sharing with low dependence on a large number of data and a tedious training process [233]. Incorporating this paradigm into our empirical experiments can allow us to provide a comprehensive comparison in terms of the number of training examples and iterations.

4.2.2 Results

- Pre-trainted Tiny-YOLOv3 vs Transfer Learning

For further comparison with our method, we explore the benefit of transfer learning in improving Tiny-YOLOv3 detection performance. Figure 4.3 shows the quantitative results of the COCO pre-trained Tiny-YOLOYv3 versus the fine-tuned model on the Caltech Pedestrian dataset, in terms of miss rate and false-positive per image (FPPI).

As shown in Figure 4.3, the transfer learning technique reduced the detection loss of the pre-trained Tiny-YOLOv3 by 9%. This observation is explained by the fact that domain adaptation was achieved by training the pre-trained model on a target fully labeled dataset. Using the visual pattern knowledge learned from the COCO dataset as the source domain, the output of the Tiny-YOLOv3 model was guided from the detection of various object classes to the precise location of pedestrian objects in the Caltech Pedestrian dataset, the target domain, with a high objectness score (from multiclass to binary object detection). Some qualitative examples of detection results on the Caltech test set, using the two Tiny-YOLOv3 models, are shown in Figures 4.4 and 4.5.

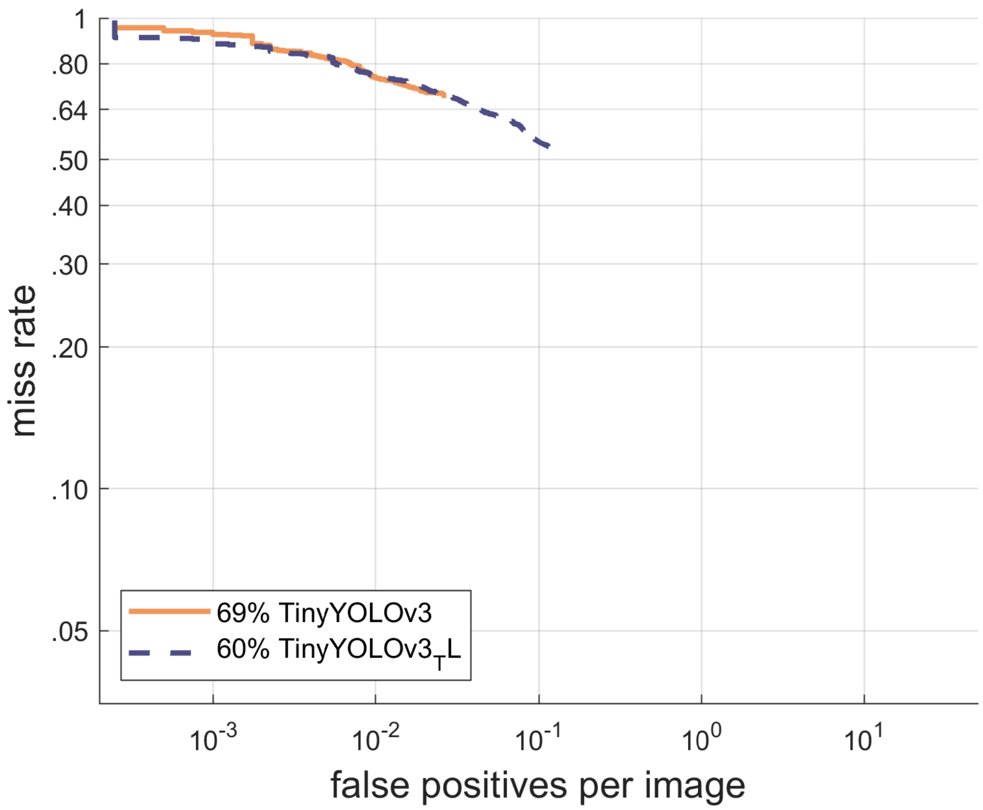


Figure 4.3: Performance curves of Caltech Pedestrian fine-tuned Tiny-YOLOv3 vs COCO pre-trained Tiny-YOLOv3

- Random Sampling

In this scenario, we investigate the random selection technique to randomly sample B instances for query manual labeling at each cycle, while setting B = 500 as indicated in [10]. TABLE 4.2 illustrates that starting from the 2^{nd} cycle, the updated model performs close to the pre-trained and fine-tuned Tiny-YOLOv3 models. However, starting from the 6^{th} cycle, the updated model clearly outperforms both previous models (57% against the 69% and 60% respectively) with only 3,000 labeled frames. This is due to more knowledge being gained from the Caltech Pedestrian dataset by the trained model as the labeled frames in U_{ann} increase.

Additionally, one can observe a varied number of detected bounding boxes from one cycle to another. This is due to the model exploiting random knowledge from the training set consisting of randomly selected informative samples.

- Experiment on WCR

To further analyze the effectiveness of our DAL algorithm, we carefully evaluated the classification and regression uncertainty-based sampling strategy for selecting a fixed B

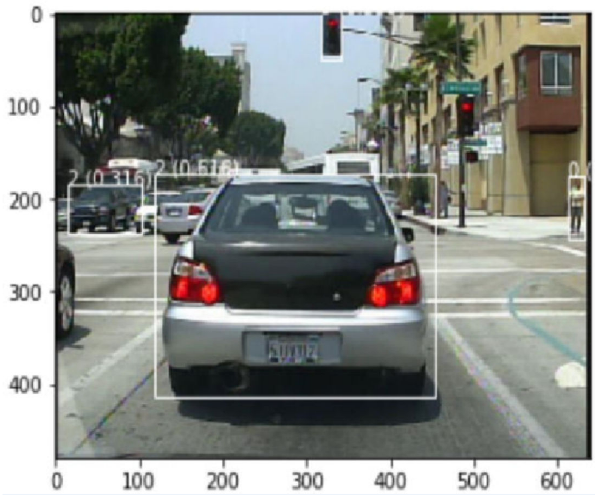


Figure 4.4: Qualitative results of the pre-trained Tiny-YOLOv3 model on the Caltech Pedestrian dataset (detect 80 object categories)

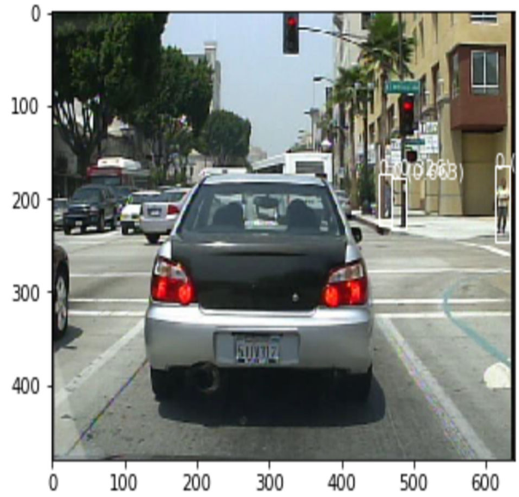


Figure 4.5: Qualitative results of the fine-tuned Tiny-YOLOv3 model on the Caltech Pedestrian dataset(detect only pedestrian object).

value, defined as in the previous experiment, of informative samples while considering an equal annotation cost for the overall unlabeled images.

- *Experiment using classification uncertainty*:

To provide a short analysis of our classification uncertainty selection strategy (U_c), we compare its overall performance, in terms of miss rate and false-positive per image (FPPI), to those of pre-trained and fine-tuned models, while evaluating three aggregation methods, namely "sum", "avg" and "max", for the earliest DAL cycles. Figures 4.6 and 4.7 together provide the quantitative results.

At the 1^{st} and 2^{nd} Uc-DAL cycles, results demonstrate the potential of the DAL strategy to yield the same or lower detection loss with only a few labeled frames. In contrast to labeled frames randomly sampled or selected by sum and max aggregation

Table 4.2: Evaluation performance results for random sampling experiment.

	cyc	#SI	#IBx	#BxC	#Bx	FPPI
RS B=500	1	500	218	486	486	85%
	2	1000	442	601	1087	69%
	3	1500	683	655	1742	62%
	4	2000	930	637	2379	62%
	5	2500	1163	547	2926	61%
	6	3000	1411	605	3531	57%
	7	3500	1637	579	4110	57%
	8	4000	1875	583	4693	55%
	9	4250	2006	294	4987	53%
PTY3						69%
TLTY3		4250	2006		4987	60%

cyc:cycles ,**#SI**:Number-selected-images ,**#IBx**:Number-images-with-Bboxes ,**#BxC**:Number-detected-Bboxes-per-cyc ,**#Bx**:Number-detected-Bboxes#

methods at the 3^{thd} cycle, the miss rate decreases to 57% when the 1500 labeled frames, selected by the "avg" method, are involved in the detector's training. This is primarily due to ignoring outliers, considered noisy samples, during the sample selection process.

By selecting a subset of the most informative samples during DAL cycles, the "avg" aggregation method can build a detector model with lower detection loss compared to transfer learning using random samples from the fully labeled Caltech Pedestrian dataset, and other methods, as shown in Figure 4.7, depicting the miss rate per cycle. However, sum and max aggregation methods might achieve the same performance but at the cost of more burden due to the outliers' influence and visual similarity between selected frames.

Overall, we can claim that the DAL algorithm based on classification uncertainty is effective in training a detection model that guarantees the expected performance with less training effort and manual labeling. Yet, this comes at the cost of negative influence from both outliers and visual similarity.

- *Experiment using regression uncertainty incorporated with classification uncertainty (WCR)*:

In this experiment, we analyze the exploration of model awareness about the class and localization prediction in addressing the aforementioned issues. The high-scoring frames are selected according to the weighted selection "WCR" (U_s) criterion based on U_c and U_r. Keeping the same fixed value of B, TABLE 4.3 to TABLE 4.5 show the results of the miss rate obtained with respect to the number of pedestrian instances (instance-level labels) selected by every aggregation method after completion of a WCR-DAL cycle.

Compared to "sum" and "max" aggregation methods, the fine-tuning of the underlying detector model using the top-ranked labeled pedestrian instances according to the "avg" aggregation method is more accurate. ≅3195 labeled boxes, collectively contained in 1500 frames selected by the "avg" method are more accurate (with 57% of detection loss) than ≅3119 selected by the "max" method (with 58% of detection loss) and ≅3088

Table 4.3: Evaluation performance results for WCR deep active learning experiment (with “sum” aggregation method)

	cyc	#SI	#IBx	#BxC	#Bx	FPPI
WCR-DAL sum	1	500	440	1764	1764	71%
	2	1000	802	862	2626	61%
	3	1500	1024	462	3088	60%
	4	**2000**	1231	415	3503	**57%**
	5	2500	1486	437	3940	58%
	6	3000	1709	518	4458	56%
	7	3500	1817	183	4641	56%
	8	4000	1872	98	4739	54%
	9	**4250**	2006	248	4987	**55%**

cyc:cycles ,**#SI**:Number-selected-images ,**#IBx**:Number-images-with-Bboxes ,**#BxC**:Number-detected-Bboxes-per-cyc ,**#Bx**:Number-detected-Bboxes#, **WCR-DAL**: WCR-DAL training, agg func= "Sum" , B=500

Table 4.4: Evaluation performance results for WCR deep active learning experiment (with “avg” aggregation method)

	cyc	#SI	#IBx	#BxC	#Bx	FPPI
WCR-DAL avg	1	500	418	1405	1405	72%
	2	1000	849	1337	2742	61%
	3	**1500**	1062	453	3195	**57%**
	4	2000	1263	428	3623	55%
	5	2500	1479	347	3970	57%
	6	3000	1652	265	4235	54%
	7	3500	1746	152	4387	57%
	8	**4000**	1863	245	4632	**51%**
	9	4250	2006	355	4987	52%

cyc:cycles ,**#SI**:Number-selected-images ,**#IBx**:Number-images-with-Bboxes ,**#BxC**:Number-detected-Bboxes-per-cyc ,**#Bx**:Number-detected-Bboxes#, **WCR-DAL**: WCR-DAL training, agg func= "Avg" , B=500

Table 4.5: Evaluation performance results for WCR deep active learning experiment (with “max” aggregation method)

	cyc	#SI	#IBx	#BxC	#Bx	FPPI
WCR-DAL max	1	500	426	1632	1632	74%
	2	1000	819	1004	2636	63%
	3	1500	1049	483	3119	58%
	4	2000	1282	506	3625	58%
	5	2500	1502	369	3994	56%
	6	3000	1631	200	4194	55%
	7	3500	1733	168	4362	55%
	8	**4000**	1874	306	4668	**54%**
	9	4250	2006	319	4987	55%

cyc:cycles ,**#SI**:Number-selected-images ,**#IBx**:Number-images-with-Bboxes ,**#BxC**:Number-detected-Bboxes-per-cyc ,**#Bx**:Number-detected-Bboxes#, **WCR-DAL**: WCR-DAL training, agg func= "Max" , B=500

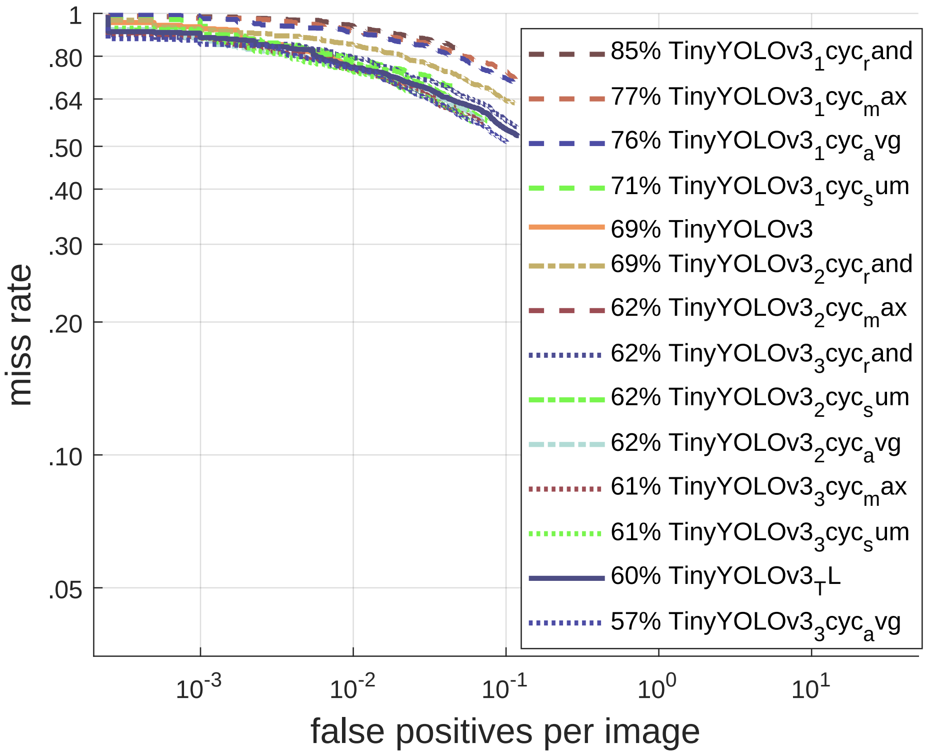


Figure 4.6: Performance curves of Pre-trained Tiny-YOLOv3 vs. TL-fine-tuned Tiny-YOLOv3 vs. random selection vs. Actively-fine-tuned Tiny-YOLOv3 (score function: Uc, aggregation method: sum, max and avg) at different training cycles on the Caltech Pedestrian dataset.

selected by "sum" method (with 60% of detection loss). This can be explained by the effectiveness of the "avg" method in avoiding outlier selection, which is the main issue of the uncertainty-based sampling strategy. As a result, the selection of frames with sparse object density can be avoided and more informative pedestrian instances can be highly ranked in hopes of rapidly reducing detection loss.

Moreover, one can note that the U_c scoring function performs slightly close to the WCR counterpart. Figure 4.6 and Figure **??** illustrate this observation by comparing the miss rate of the three aggregation methods. This is primarily due to the failure of the sampling strategy to capture the visual patterns' similarity in subsequent frames.

- Experiment on CEDBMAL

In this experiment, we analyze the importance of involving a dynamic batch selection to address the variable annotation cost issue and improve the performance. To this end, a group of frames, with the best batch size B, is sampled according to the labeling time cost of frames and the distribution of objects over them.

TABLE 4.6 shows the results of the miss rate according to the “avg” aggregation function and the best B value selected at each CEDBMAL cycle. It is observed that the miss rate, in the 2^{nd} cycle, is decreased to 57% with only 783 labeled frames that contain

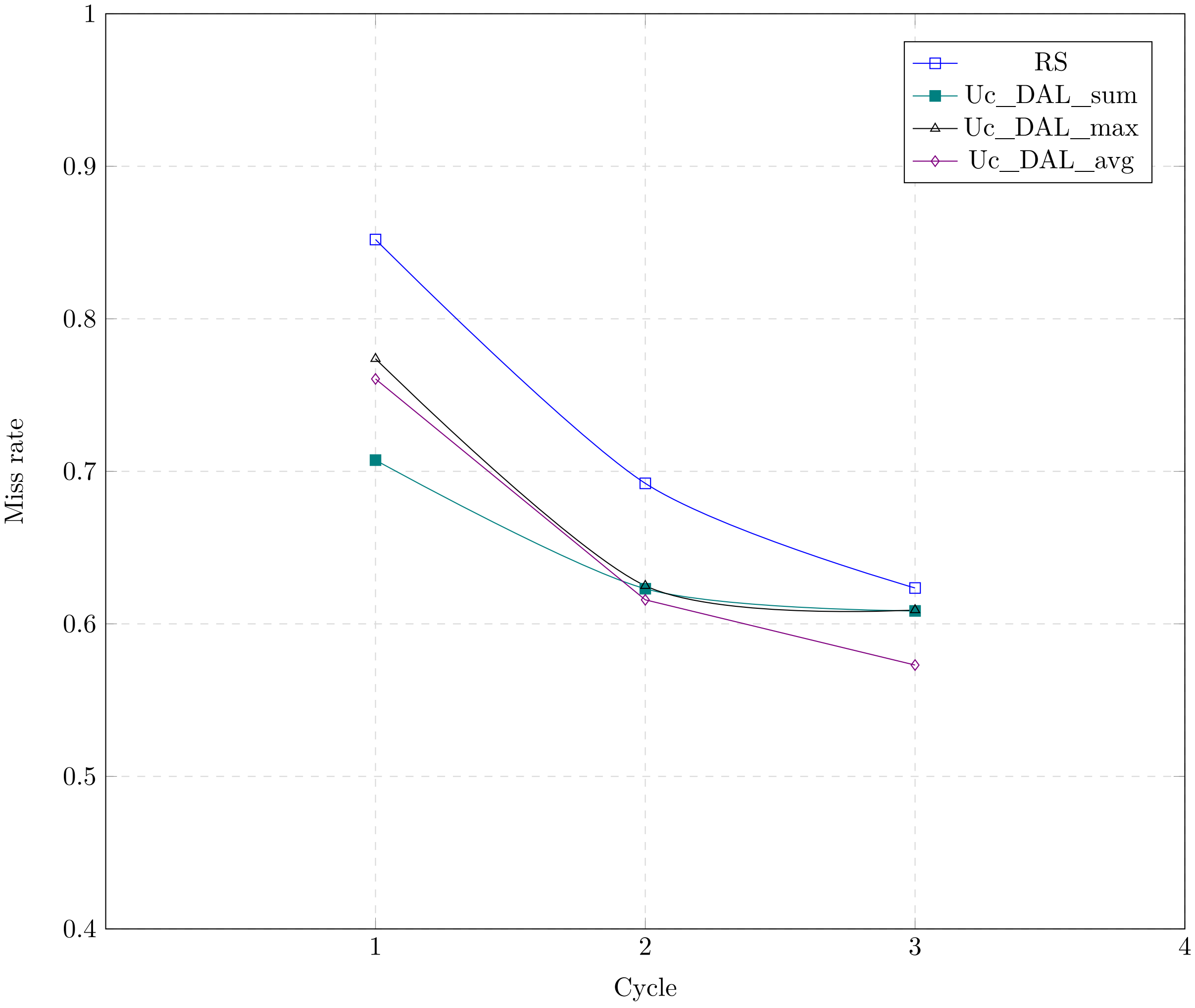


Figure 4.7: Miss rate of random selection vs. variants of our UC_DAL method based on "sum", "max" and "avg" aggregation method.

≅ 2226 pedestrians. The same miss rate is obtained using random sampling method after 6 cycles (3000 selected frames for labeling containing ≅ 3531 pedestrians as reported in TABLE 4.2), and using WCR-DAL method in 3^{thd} cycle, but at the cost of more labeled frames (1062 frames which contain ≅ 3195 pedestrians) and a fixed group size (see TABLE 4.4). Such observation is explained by two reasons: (1) the picking up, in a cost-aware manner, of the best group with few important frames that contain relatively diverse and fewer (but more informative) detected pedestrians. (2) The integration of WCR uncertainty to estimate pedestrian objects amount during batch sampling and optimal batch size selection.

- Comparisons with state-of-the-art approaches

In the following parts, we undertake a comparative analysis of the detection performance of our proposed method with that of existing passively trained baseline

Table 4.6: Evaluation performance results for CEDBMAL experiment (with "avg" as aggregation method)

	cyc	#B	#IBx	#BxC	#Bx	FPPI
CEDBMAL	0	500	218	486	486	85%
	1	900	707	1507	1993	61%
	2	100	**783**	233	2226	**57%**
	3	500	1025	670	2896	55%
	4	500	1276	632	3528	56%
	5	100	1399	159	3687	57%
	6	500	1501	364	4051	57%
	7	100	1600	147	4198	55%
	8	500	1773	385	4583	55%
	9	100	1828	108	4691	55%
	10	100	1909	104	4795	55%
	11	350	2006	192	4987	**53%**

cyc:cycles ,**#B**:Best-batch-size ,**#IBx**:Number-images-with-Bboxes ,**#BxC**:Number-detected-Bboxes-per-cyc ,**#Bx**:Number-detected-Bboxes#, **CEDBMAL**: CEDBMAL training, agg func= "Avg"

pedestrian detectors. We then proceed to compare it with existing DAL techniques in the related literature for training pedestrian detectors.

- *Comparisons with baseline pedestrian detector*:

In this part, we compare the results of using our DAL strategy versus standard training strategies for building pedestrian detectors. The experiment is conducted using representative shallow learning (handcrafted feature)-based and deep (feature) learning-based pedestrian detectors whose results are published on Caltech Pedestrian detection benchmark [234, 235]. All methods considered here were trained on fully labeled Caltech-USA and INRIA datasets without referring to DAL algorithms.

Table 4.7: Listing of methods for pedestrian detection considered in comparison on Caltech-USA dataset

method	Td	Ts	Fe	Cl	#LtD	#Dp	MR
ConvNet [236]	INRIA	Sot	learning(Pixels)	DeepNet	21845	-	0.77
TinyYOLOv3_WCRDAL_1avg(ours)	Caltech	AL + TL	learning(Pixels)	DeepNet	418	1405	0.72
TinyYOLOv3	COCO	Pre	learning(Pixels)	DeepNet	165482	-	0.69
HOG [159]	INRIA	Scratch	handcrafted	LinearSVM	≅ 14658	-	0.68
MLS [237]	INRIA	boosting	handcrafted	Adaboost	≅ 14658	-	0.61
TinyYOLOv3_WCRDAL_2avg(ours)	Caltech	AL + TL	learning(Pixels)	DeepNet	849	2742	0.61
TinyYOLOv3_CEDBMAL_1(ours)	Caltech	AL + TL	learning(Pixels)	DeepNet	707	1993	0.61
TinyYOLOv3_TL	Caltech	TL	learning(Pixels)	DeepNet	2006	-	0.59
TinyYOLOv3_WCRDAL_3avg(ours)	Caltech	AL + TL	learning(Pixels)	DeepNet	1062	3195	0.57
TinyYOLOv3_CEDBMAL_2(ours)	Caltech	AL + TL	learning(Pixels)	DeepNet	783	2226	0.57
TinyYOLOv3_CEDBMAL_3(ours)	Caltech	AL + TL	learning(Pixels)	DeepNet	1025	2896	0.55
Katamari [234]	Caltech	scratch	handcrafted	-	-	-	0.22
TLL-TFA [238]	Caltech	scratch	learning(Pixels)	DeepNet	≅ 42782	-	0.07

Td:Training-dataset ,**Ts**:Training-strategy ,**Fe**:Features ,**Cl**:Classifier ,**#Ltd**:Number-labeled-training-data ,**#Dp**:Number-detected-pedestrian ,**MR**:Miss-Rate ,**Sot**:Stochastic online training ,**TL**:Transfer learning ,**AL**:Active learning ,**Pre**:Pre-trained ,**CEDBMAL_x**: CEDBMAL training, agg func= "Avg", cycle number= "x" ,**WCRDAL_xavg**: WCRDAL training, agg func= "Avg", cycle number= "x"

Figure 4.8 provides quantitative results in terms of miss rate and false-positive per image (FPPI). The results depict that the deep-learned features train a more accurate

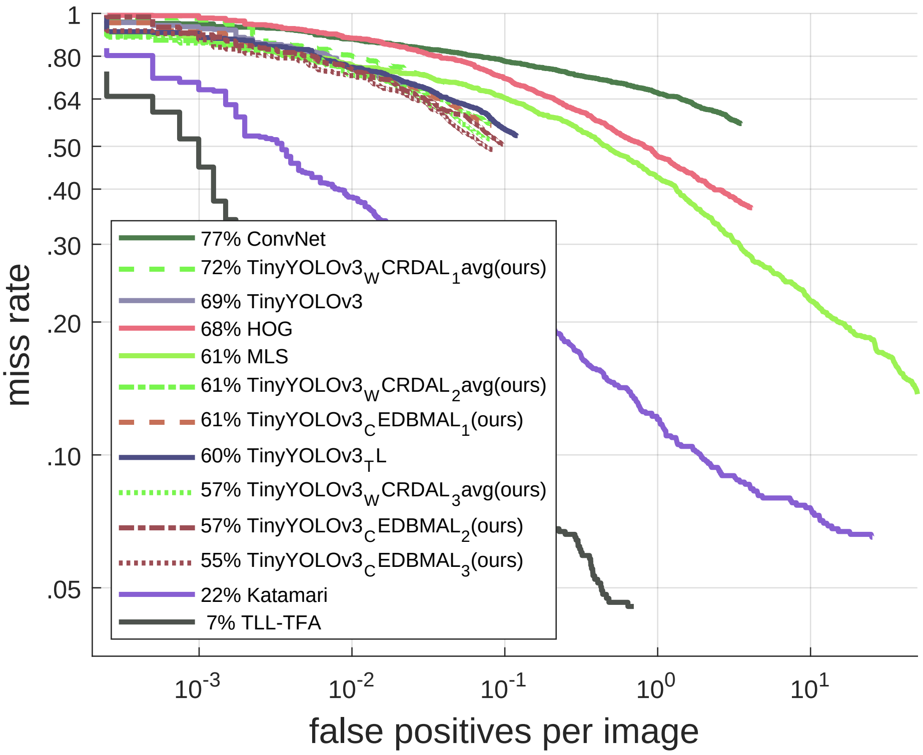


Figure 4.8: Pedestrian detection on the Caltech Pedestrian dataset.

pedestrian detector than handcrafted features. This is due to the fact of the model's sensitivity towards the training strategy and the amount of data used for knowledge learning.

Moreover, the results report that the subset of labeled training data, actively selected and accumulated by our proposed methods, is enough to yield the best performance and outperforms some pedestrian detectors with more than 14% reduction in miss rate in the early DAL cycles (57% MR of CEDBMAL with 2226 labeled pedestrian objects against 57% MR of WCR-DAL with 3503 labeled pedestrian objects against 77% MR of ConvNet with fully labeled dataset).

Beyond labeling cost awareness, we can claim that using batch mode DAL in conjunction with TL could lead to efficiently training deep learning-based approaches with less amount of training data, less architecture complexity, and an attenuated negative impact of outlier, redundancy data, and domain shift issues.

TABLE 4.7 reports additional details on the training data and the miss rate versus the labeled training data as well as the number of detected pedestrians.

- *Comparisons with DAL-based pedestrian detector*:

In this part, we evaluate our method compared to published results of the related DAL technique [10] for training pedestrian detectors, while performing per-cycle comparisons. In this comparative analysis, we examine the miss rate by considering the impact of both the number of detected pedestrians and the batch size. Overall, the comparison settings are summarized in TABLE 4.8. Quantitative results are reported in Figure 4.9 together with Figure 4.10.

As shown, the results indicate that the three DAL-based methods outperform the random sampling strategy. Since the first DAL cycle can mine hard instances, it contributes the most in terms of reducing the miss rate compared to randomly sampled instances (see Figure 4.9).

Table 4.8: Comparison settings of our method to Aghdam et al.[10] .

method	Td	OD	Qs	#Cyc	Bs	B	Sf	Af
Aghdam et al.[10]	CP, C, BD	dDNa	oBo	14	Fi	500	pSf, MC-D, En	dAf
WCRDAL(ours)	C	TYv3	oBo	9	Fi	500	Un	avg, max, sum
CEDBMAL(ours)	C	TYv3	Bat	11	Dy	Dy	Un and Di	avg

Td:Training-dataset ,**OD**:Object-detector ,**QS**:Query-strategy ,**#Cyc**:DAL-cycles-number ,**Bs**:Batch-size-selection ,**B**:Batch-size ,**Sf**:Scoring-functuin ,**Af**:Aggregation-function ,**dDNa**:defined-Deep-Network-architecture ,**dAf**:defined-Aggregation-function ,**MC-D**:Monte Carlo-Dropout ,**Fi**:fixed (static) ,**pSf**:pixel-level Sf ,**En**:entropy ,**oBo**:one-by-one query method ,**CP**:CityPerson Pedestrian ,**BD**:BDD100K ,**Un**:Uncertainty ,**Di**:Diversity ,**Bat**:batch query method ,**CP**:TinyYOLOv3 ,**TYv3**:TinyYOLOv3 ,**Dy**: dynamic ,**C**: Caltech ,**CEDBMAL**: CEDBMAL training

Nevertheless, at the end of 3^{thd} cycle, the network trained on the 1500 frames selected by the method [10] is more accurate than the networks trained on the same number of selected frames by our DAL methods. From Figure 4.9 and Figure 4.10, we can see a reduction of miss rate about 15% under the cost of labeling ≅ 1900 predicted pedestrian instances against 15% with the cost of labeling ≅ 3195 and about 20% for a labeling cost of 2K predicted pedestrian instances. This is due to the ability of the method [10] to query the labeling of the most useful detected pedestrian instances, providing the network with more knowledge about the target object.

Compared to per-instance sampling strategies, our proposed adaptive batch query strategy performs better than our WCR-DAL method and exhibits a performance close to the method reported in [10]. This observation is emphasized by the gains in handling the bounding box distribution, across DAL cycles, regardless of the underlying detector architecture. Consequently, CEDBMAL's dynamic selection of batch size based on object amount not only helps to effectively maintain the cost of data labeling but also reduces DAL selection cycles and naturally supports the commonly used mini-batch training concept.

4.2.3 Discussion

As can be seen, the reported results on the Caltech Pedestrian dataset are very promising. However, the transfer learning presented a worse performance. Such observation can be explained by the fact that considering outliers and redundant data during the training process degraded the detector model performance. The random sampling selection technique, coupled with transfer learning, surpasses the transfer learning method because

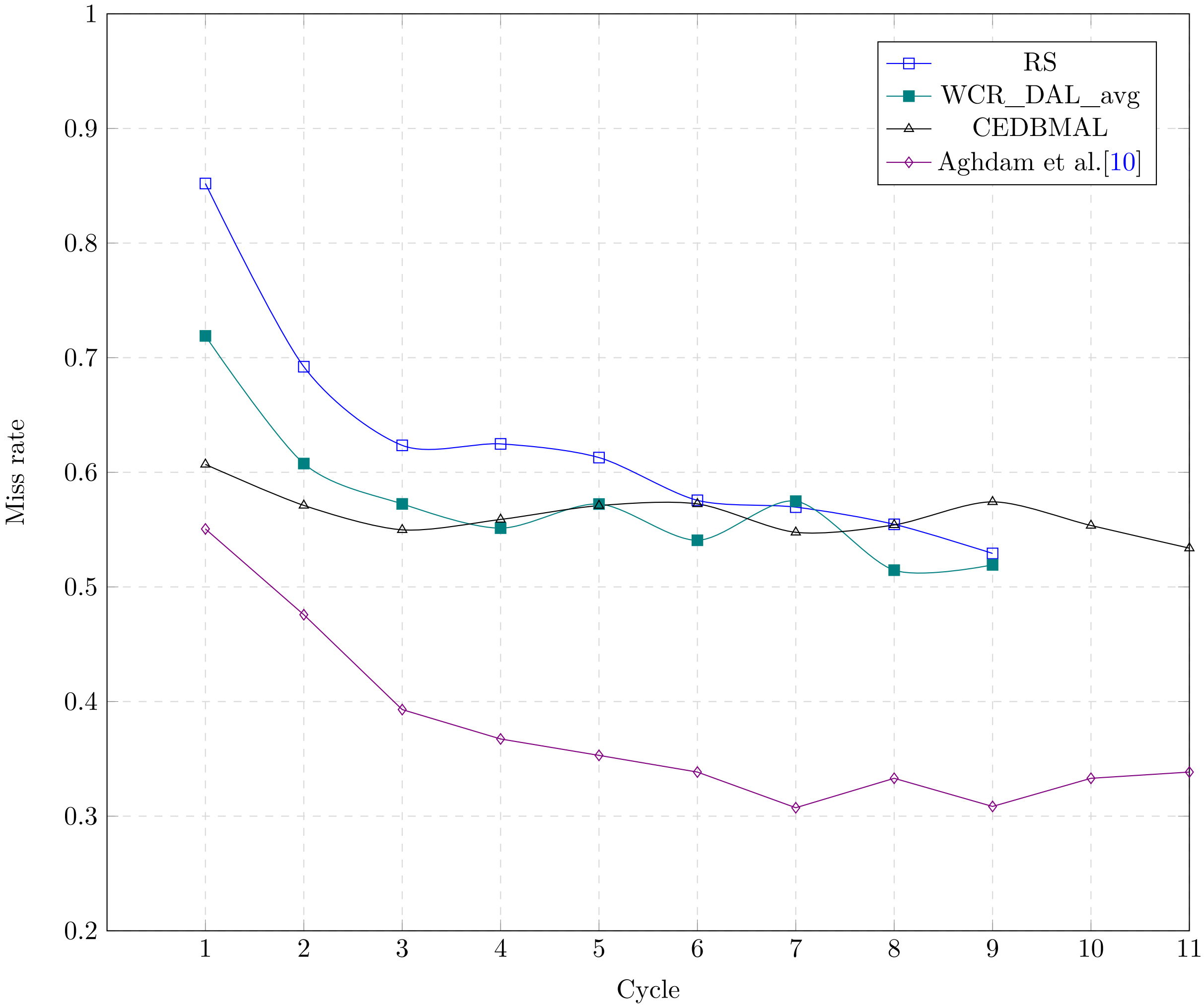


Figure 4.9: Miss rate of random selection vs. WCR__DAL__avg (ours) vs. CEDBMAL (ours) vs. Aghdam et al.[10] .

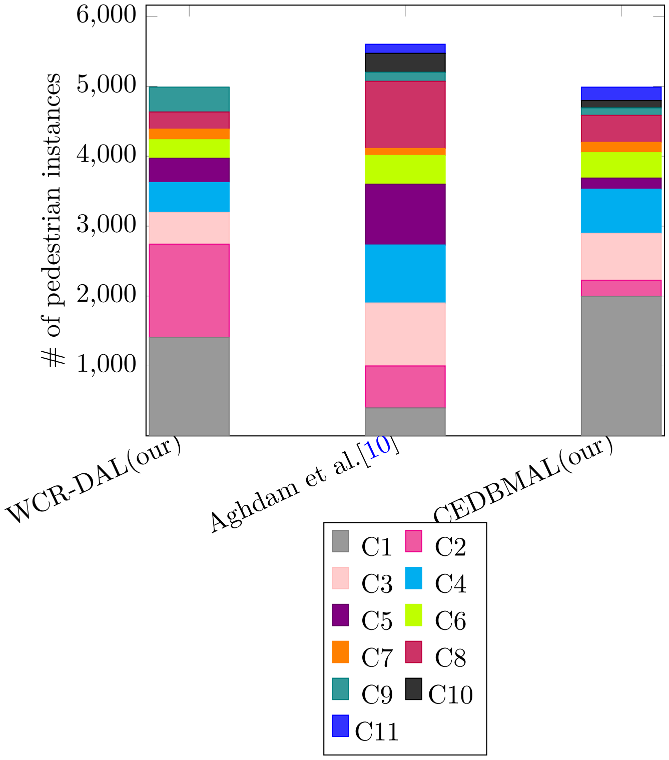


Figure 4.10: Number of pedestrian instances in training set at each DAL cycle

it does not suffer from these issues. Compared to previous methods, our proposed DAL method effectively decreases the detection loss while minimizing annotation and training costs and dealing with the negative influence of noisy training data.

Regardless of the aggregation method used, both U_c and WCR query strategies can gradually discover more knowledge from a few frames, leading to min more informative boxes (hard examples) that provide a good signal for fast convergence and annotation cost reduction, while the overall performance of both strategies remains close to each other. Even though WCR-DAL could select high uncertainty frames with more pedestrian objects in early cycles, similar object distribution in consecutive frames does not always yield an improvement and yet decreases the detector's performance. Throughout the adaptive selection of the best batch according to its size, target object distribution, and annotation cost, it is clear that CEDBMAL cost-effectively fine-tunes a more robust CNN-based detection model and conserves detection loss close to existing performance results. This is due to maintaining outliers' selection and diversity between selected frames, which is highly expected to decrease the detection loss while saving annotation time within a given budget. However, the success of our method is a matter of critical factors, namely the underlying detector architecture complexity, scoring functions, and the query strategy.

Although existing object detection algorithms have achieved good results, it remains challenging to effectively handle the correlation between sample selection criteria, dynamic batch selection, and noisy data identification to minimize the cost of data labeling.

Conclusion

This chapter has introduced and explained our Cost-Effective Deep Batch Mode Active Learning (CEDBMAL) framework, an efficient labeling and training pipeline for building accurate and robust deep object detection model for real-world deployment in autonomous vehicles. We initially presented the primary components of this framework, including the detection model and the defined DAL approach dedicated to CNN-based object detectors. This approach consists of two contributions: uniform-cost and batch-aware query strategies. The uniform-cost DAL approach combines various criteria in the query strategy, such as combined classification and regression uncertainty, and diversity. However, batch-aware DAL provides an additional boost under labeling time constraints compared to other DAL approaches, leveraging optimization techniques to query the label of an optimal batch with more informative, less noisy, and less redundant frames. Subsequently, we provided experimental results comparing our contributions to random sampling, transfer learning, and uniform-cost AL approaches using various evaluation criteria. Finally, we concluded by discussing the findings and limitations.

The findings demonstrated that the CEDBMAL framework outperforms the most commonly used learning schemes and efficiently addresses the visual similarity and domain shift restrictions in public autonomous driving datasets. Additionally, it limits the number of queries to adhere to annotation budget and training resource constraints, dynamically adapting at the frame (instance) level while considering the number of valuable on-road objects remaining in the unlabeled pool that the model is uncertain about.

Conclusion and future research directions

This chapter provides a broad summary of the thesis as well as a few suggestions for further investigation.

4.3 Conclusion

Automated driving systems could greatly enhance safety and mobility for all travelers. Ensuring driving safety in the open and complex environments faced by autonomous vehicles demands highly robust detection algorithms. Current research focuses on exploring the latest advancements in deep learning to develop high-performance object detection models. Most existing deep object detectors in autonomous driving rely on fully annotated public benchmarks and passive supervised learning algorithms. However, a few employ deep active learning schemes, recognized as efficient approaches to overcome challenges in data labeling and model training, thereby achieving significant improvements in robust and accurate object detection within this domain.

This doctoral thesis is dedicated to developing deep active learning solutions aimed at enhancing object detection in autonomous driving. To achieve this goal, we propose using uncertainty heuristics as a metric for on-road object detection. Based on this criterion, a batch-based query strategy is proposed to sequentially capture top-ranked samples. This strategy operates within a uniform-cost Deep Active Learning (DAL) framework. However, in autonomous driving, datasets can vary significantly in terms of data modalities, environmental scenarios, object appearances, and labeling costs.

Our second contribution, the Cost-Effective Deep Batch Mode Active Learning (CEDBMAL) framework, addresses these challenges. This framework ensures efficient labeling and training processes without imposing additional burdens in terms of human annotator intervention, DAL iterations, or exceeding annotation budget limitations.

To validate our proposals, various experiments were conducted on the Caltech Pedestrian dataset, demonstrating that uniform-cost DAL and CEDBMAL can significantly reduce labeling costs in autonomous driving and achieve high-performance pedestrian detection in terms of miss rate metric and number of pedestrian detection at each DAL cycle. Notably, experiment findings demonstrate that uniform-cost DAL outperforms random sampling and transfer learning in terms of miss rate with fewer labeled images. Furthermore, experiments conducted on CEDBMAL have demonstrated an enhancement in uniform-cost DAL's ability to achieve a miss rate comparable to that of a generic deep pedestrian detector trained passively. Additionally, CEDBMAL increased the number of detected pedestrians during the early DAL cycle compared to other uniform-cost DAL methods using a specific deep pedestrian detector.

Some future research directions and perspectives are highlighted below.

4.4 Future works

Several future directions for deep active learning research have to be considered to ensure the successful application of object detection in autonomous vehicles. These include:

- **Efficient query strategies**: Develop more efficient and effective query strategies for selecting informative samples in deep active learning. This includes exploring novel uncertainty measures, diversity-based sampling techniques, and strategies such as consistency for handling class imbalance and rare objects.

- **Non-strongly supervised learning**: To reduce strong dependency on manually labeled data, researchers can investigate integrating non-strongly supervised learning schemes, such as self-supervised, semi-supervised, and others, with active learning. Self-supervised learning can generate pseudo-labels or auxiliary tasks to pre-train models on large amounts of unlabeled data, which can then be fine-tuned with active learning using smaller labeled datasets.

  Real-time performance measurement: A promising avenue for future research is optimizing DL-based object detection models for real-time, latency-sensitive applications, ensuring they meet the desired performance requirements, such as low latency balanced with accuracy. This is especially important when deploying models over public benchmarks in resource-constrained, low-latency autonomous driving environments. Measuring latency performance in the context of active learning involves evaluating how quickly an object detection model can make predictions after being trained on a new batch of labeled data and how efficiently it incorporates newly labeled samples into its training process. This requires combining traditional latency metrics, such as inference latency, training latency, model convergence speed, and throughput, with specific active learning metrics, such as verification latency, query latency, and active learning cycle efficiency, while also exploring techniques to reduce inference time without compromising accuracy.

Bibliography

[1] Y.-Q. Huang, J.-C. Zheng, S.-D. Sun, C.-F. Yang, and J. Liu, "Optimized yolov3 algorithm and its application in traffic flow detections," *Applied Sciences*, vol. 10, no. 9, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/9/3079

[2] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611–629, 2018.

[3] E. Al Hadhrami, M. Al Mufti, B. Taha, and N. Werghi, "Transfer learning with convolutional neural networks for moving target classification with micro-doppler radar spectrograms," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 2018, pp. 148–154.

[4] K. Grauman and B. Leibe, *Visual object recognition*. Morgan & Claypool Publishers, 2011, no. 11.

[5] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, pp. 261–318, 2020.

[6] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: A review," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 5, pp. 694–711, 2006.

[7] J. Xu, Y. Huang, and D. Ying, "Traffic sign detection and recognition using multi-frame embedding of video-log images," *Remote Sensing*, vol. 15, no. 12, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/12/2959

[8] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[9] ——, "From theories to queries: Active learning in practice," in *Active learning and experimental design workshop in conjunction with AISTATS 2010*. JMLR Workshop and Conference Proceedings, 2011, pp. 1–18.

[10] H. H. Aghdam, A. Gonzalez-Garcia, J. v. d. Weijer, and A. M. López, "Active learning for deep detection neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2019, pp. 3672–3680.

[11] E. Khatab, A. Onsy, M. Varley, and A. Abouelfarag, "Vulnerable objects detection for autonomous driving: A review," *Integration*, vol. 78, pp. 36–48, 2021.

[12] Z. Li, Y. Du, M. Zhu, S. Zhou, and L. Zhang, "A survey of 3d object detection algorithms for intelligent vehicles development," *Artificial Life and Robotics*, pp. 1–8, 2022.

[13] A. Ghasemieh and R. Kashef, "3d object detection for autonomous driving: Methods, models, sensors, data, and challenges," *Transportation Engineering*, vol. 8, p. 100115, 2022.

[14] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[15] R. Naja *et al.*, *Wireless vehicular networks for car collision avoidance*. Springer, 2013, vol. 2013.

[16] A. Aldakkhelallah and M. Simic, "Autonomous vehicles in intelligent transportation systems," in *Human Centred Intelligent Systems: Proceedings of KES-HCIS 2021 Conference*. Springer, 2021, pp. 185–198.

[17] J. Wang, L. Zhang, Y. Huang, J. Zhao, and F. Bella, "Safety of autonomous vehicles," *Journal of advanced transportation*, vol. 2020, pp. 1–13, 2020.

[18] O. Tengilimoglu, O. Carsten, and Z. Wadud, “Implications of automated vehicles for physical road environment: A comprehensive review,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 169, p. 102989, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554522003660

[19] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos, “An overview of autonomous vehicles sensors and their vulnerability to weather conditions,” *Sensors*, vol. 21, no. 16, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/16/5397

[20] T. Miqdady, R. de Oña, J. Casas, and J. de Oña, “Studying traffic safety during the transition period between manual driving and autonomous driving: A simulation-based approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6690–6710, 2023.

[21] “Comprehensive safety assessment in mixed fleets with connected and automated vehicles: A crash severity and rate evaluation of conventional vehicles,” *Accident Analysis Prevention*, vol. 142, p. 105567, 2020.

[22] Q. Rao and J. Frtunikj, “Deep learning for self-driving cars: Chances and challenges,” in *Proceedings of the 1st international workshop on software engineering for AI in autonomous systems*, 2018, pp. 35–38.

[23] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, “Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues,” *Array*, vol. 10, p. 100057, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005621000059

[24] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, “Vehicular ad hoc networks (vanets): status, results, and challenges,” *telecommunication systems*, vol. 50, pp. 217–241, 2012.

[25] N. H. Hussein, C. T. Yaw, S. P. Koh, S. K. Tiong, and K. H. Chong, “A comprehensive survey on vehicular networking: Communications, applications, challenges, and upcoming research directions,” *IEEE Access*, vol. 10, pp. 86 127–86 180, 2022.

[26] S. Zeadally, M. A. Javed, and E. B. Hamida, “Vehicular communications for its: Standardization and challenges,” *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 11–17, 2020.

[27] K.-C. Su, H.-M. Wu, W.-L. Chang, and Y.-H. Chou, “Vehicle-to-vehicle communication system through wi-fi network using android smartphone,” in *2012 International conference on connected vehicles and expo (ICCVE)*. IEEE, 2012, pp. 191–196.

[28] R. Hussain, J. Lee, and S. Zeadally, “Trust in vanet: A survey of current solutions and future research opportunities,” *IEEE transactions on intelligent transportation systems*, vol. 22, no. 5, pp. 2553–2571, 2020.

[29] S. Zalte, V. Ghorpade, and R. K. Kamat, “Synergizing blockchain, iot, and ai with vanet for intelligent transport solutions,” *Emerging Computing Paradigms: Principles, Advances and Applications*, pp. 193–210, 2022.

[30] M. M. Rana and K. Hossain, “Connected and autonomous vehicles and infrastructures: A literature review,” *International Journal of Pavement Research and Technology*, pp. 1–21, 2021.

[31] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, “Sensor technologies for intelligent transportation systems,” *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[32] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil, “How to build vehicular ad-hoc networks on smartphones,” *Journal of Systems Architecture*, vol. 59, no. 10, pp. 996–1004, 2013.

[33] T. De Borba, O. Vaculín, H. Marzbani, and R. N. Jazar, “Increasing safety of automated driving by infrastructure-based sensors,” *IEEE Access*, 2023.

[34] M. Tsukada, T. Oi, M. Kitazawa, and H. Esaki, “Networked roadside perception units for autonomous driving,” *Sensors*, vol. 20, no. 18, p. 5320, 2020.

[35] M. Williams, “Prometheus-the european research programme for optimising the road transport system in europe,” in *IEE Colloquium on Driver Information*. IET, 1988, pp. 1–1.

[36] B. Ulmer, “Vita ii-active collision avoidance in real traffic,” in *Proceedings of the Intelligent Vehicles’ 94 Symposium*. IEEE, 1994, pp. 1–6.

[37] Z. Chen, “Computer vision and machine learning for autonomous vehicles,” *View at*, 2017.

[38] N. H. T. S. A. United States Department of Transportation. (2019) Overview of automated vehicle technology. [Online]. Available: https://www.nhtsa.gov/document/overview-automated-vehicle-technology

[39] T. Litman, “Autonomous vehicle implementation predictions: Implications for transport planning,” 2020.

[40] F. Flemisch, F. Nashashibi, N. Rauch, A. Schieben, S. Glaser, G. Temme, P. Resende, B. Vanholme, C. Löper, G. Thomaidis *et al.*, “Towards highly automated driving: Intermediate report on the haveit-joint system,” in *3rd European Road Transport Research Arena, TRA 2010*, 2010.

[41] Á. Takács, I. Rudas, D. Bösl, and T. Haidegger, “Highly automated vehicles and self-driving cars [industry tutorial],” *IEEE Robotics & Automation Magazine*, vol. 25, no. 4, pp. 106–112, 2018.

[42] Z. Nikolić, “Embedded vision in advanced driver assistance systems,” in *Advances in embedded computer vision*. Springer, 2014, pp. 45–69.

[43] J. Langheim, A. Buchanan, T. Automotive, and U. Lages, “Carsense-new environment sensing for advanced driver assistance systems,” *International Conference of Intelligent Transportation Systems (ITSC 2001)*, vol. 3, 01 2001.

[44] S. TSUGAWA, “Issues and recent trends in vehicle safety communication systems,” *IATSS Research*, vol. 29, no. 1, pp. 7–15, 2005.

[45] S. Krishnarao, H.-C. Wang, A. Sharma, and M. Iqbal, “Enhancement of advanced driver assistance system (adas) using machine learning,” in *International Congress on Information and Communication Technology*. Springer, 2020, pp. 139–146.

[46] J. E. Ball and B. Tang, “Machine learning and embedded computing in advanced driver assistance systems (adas),” p. 748, 2019.

[47] A. M. Meroth, F. Trankle, B. F. Richter, M. Wagner, M. Neher, and J. Luling, “Functional safety and development process capability for intelligent transportation systems,” *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 4, pp. 12–23, 2015.

[48] J. Ayoub, F. Zhou, S. Bao, and X. J. Yang, “From manual driving to automated driving: A review of 10 years of autoui,” in *Proceedings of the 11th international conference on automotive user interfaces and interactive vehicular applications*, 2019, pp. 70–90.

[49] N. H. T. S. A. United States Department of Transportation. Automated driving systems. [Online]. Available: https://www.nhtsa.gov/vehicle-manufacturers/automated-driving-systems

[50] J. Becker, M. Helmle, and O. Pink, *System Architecture and Safety Requirements for Automated Driving*. Cham: Springer International Publishing, 2017, pp. 265–283.

[51] W. Zong, C. Zhang, Z. Wang, J. Zhu, and Q. Chen, “Architecture design and implementation of an autonomous vehicle,” *IEEE access*, vol. 6, pp. 21 956–21 970, 2018.

[52] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, “Vehicular communication networks in the automated driving era,” *IEEE Communications Magazine*, vol. 56, no. 9, pp. 26–32, 2018.

[53] G. Qiaochu, “Software system of autonomous vehicles: Architecture, network and os,” Ph.D. dissertation, University of Pennsylvania, School of Engineering and Applied Science, 2020.

[54] S. Kumar, S. Gollakota, and D. Katabi, “A cloud-assisted design for autonomous driving,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 41–46.

[55] A. Mokhtarian, A. Kampmann, M. Lueer, S. Kowalewski, and B. Alrifaee, “A cloud architecture for networked and autonomous vehicles∗∗this research is accomplished within the project "unicaragil" (fkz em2adis002). we acknowledge the financial support for the project by the federal ministry of education and research of germany (bmbf).” *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 233–239, 2021, 16th IFAC Symposium on Control in Transportation Systems CTS 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589632100464X

[56] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, and A. Mouzakitis, “A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6206–6221, 2022.

[57] D. Dey, A. Habibovic, A. Löcken, P. Wintersberger, B. Pfleging, A. Riener, M. Martens, and J. Terken, “Taming the ehmi jungle: A classification taxonomy to guide, compare, and assess the design principles of automated vehicles’ external human-machine interfaces,” *Transportation Research Interdisciplinary Perspectives*, vol. 7, p. 100174, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590198220300853

[58] S. Ballingall, M. Sarvi, and P. Sweatman, “Safety assurance concepts for automated driving systems,” *SAE International Journal of Advances and Current Practices in Mobility*, vol. 2, no. 3, pp. 1528–1537, apr 2020. [Online]. Available: https://doi.org/10.4271/2020-01-0727

[59] N. Rajabli, F. Flammini, R. Nardone, and V. Vittorini, “Software verification and validation of safe autonomous cars: A systematic literature review,” *IEEE Access*, vol. 9, pp. 4797–4819, 2020.

[60] S. Behere and M. Törngren, “A functional architecture for autonomous driving,” in *Proceedings of the First International Workshop on Automotive Software Architecture*, 2015, pp. 3–10.

[61] A. García, D. Llopis-Castelló, and F. J. Camacho-Torregrosa, “From the vehicle-based concept of operational design domain to the road-based concept of operational road section,” *Frontiers in Built Environment*, vol. 8, p. 901840, 2022.

[62] M. Schratter, M. Hartmann, and D. Watzenig, “Pedestrian collision avoidance system for autonomous vehicles,” *SAE International Journal of Connected and Automated Vehicles*, vol. 2, no. 12-02-04-0021, pp. 279–293, 2019.

[63] Y. Ma, Z. Wang, H. Yang, and L. Yang, “Artificial intelligence applications in the development of autonomous vehicles: A survey,” *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.

[64] M. Alawadhi, J. Almazrouie, M. Kamil, and K. A. Khalil, “A systematic literature review of the factors influencing the adoption of autonomous driving,” *International Journal of System Assurance Engineering and Management*, vol. 11, pp. 1065–1082, 2020.

[65] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, “A survey of deep learning applications to autonomous vehicle control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2020.

[66] W. Wang, L. Wang, C. Zhang, C. Liu, L. Sun *et al.*, “Social interactions for autonomous driving: A review and perspectives,” *Foundations and Trends® in Robotics*, vol. 10, no. 3-4, pp. 198–376, 2022.

[67] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G. P. Joshi, and W. Cho, “A review on autonomous vehicles: Progress, methods and challenges,” *Electronics*, vol. 11, no. 14, p. 2162, 2022.

[68] X. Song, H. Gao, T. Ding, Y. Gu, J. Liu, and K. Tian, “A review of the motion planning and control methods for automated vehicles,” *Sensors*, vol. 23, no. 13, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/13/6140

[69] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, “Deep reinforcement learning for autonomous driving: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[70] K. Wong, Y. Gu, and S. Kamijo, “Mapping for autonomous driving: Opportunities and challenges,” *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 1, pp. 91–106, 2020.

[71] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, “Deep learning for safe autonomous driving: Current challenges and future directions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[72] R. Deng, B. Di, and L. Song, "Cooperative collision avoidance for overtaking maneuvers in cellular v2x-based autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4434–4446, 2019.

[73] O. Willers, S. Sudholt, S. Raafatnia, and S. Abrecht, "Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39.* Springer, 2020, pp. 336–350.

[74] A. Li, L. Sun, W. Zhan, M. Tomizuka, and M. Chen, "Prediction-based reachability for collision avoidance in autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 7908–7914.

[75] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A survey of driving safety with sensing, vehicular communications, and artificial intelligence-based collision avoidance," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 7, pp. 6142–6163, 2021.

[76] S. B. Prathiba, G. Raja, and N. Kumar, "Intelligent cooperative collision avoidance at overtaking and lane changing maneuver in 6g-v2x communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 112–122, 2021.

[77] H. Li, T. Zheng, F. Xia, L. Gao, Q. Ye, and Z. Guo, "Emergency collision avoidance strategy for autonomous vehicles based on steering and differential braking," *Scientific Reports*, vol. 12, no. 1, p. 22647, 2022.

[78] J. He, K. Yang, and H.-H. Chen, "6g cellular networks and connected autonomous vehicles," *IEEE Network*, vol. 35, no. 4, pp. 255–261, 2020.

[79] B. Yang, X. Cao, K. Xiong, C. Yuen, Y. L. Guan, S. Leng, L. Qian, and Z. Han, "Edge intelligence for autonomous driving in 6g wireless system: Design challenges and solutions," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 40–47, 2021.

[80] S. Hakak, T. R. Gadekallu, P. K. R. Maddikunta, S. P. Ramu, M. Parimala, C. De Alwis, and M. Liyanage, "Autonomous vehicles in 5g and beyond: A survey," *Vehicular Communications*, p. 100551, 2022.

[81] S. U. Hussain, "Machine learning methods for visual object detection," Ph.D. dissertation, Université de Grenoble, 2011.

[82] H. Wang, G. Zhao, and J. Yuan, "Visual pattern discovery in image and video data: a brief survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 1, pp. 24–37, 2014.

[83] Y. Li, S. Wang, Q. Tian, and X. Ding, "Feature representation for statistical-learning-based object detection: A review," *Pattern Recognition*, vol. 48, no. 11, pp. 3542–3559, 2015.

[84] V. N. Kemajou, A. Bao, and O. Germain, "Wellbore schematics to structured data using artificial intelligence tools," in *Offshore Technology Conference.* OTC, 2019, p. D011S010R006.

[85] W. Chen, Y. Li, Z. Tian, and F. Zhang, "2d and 3d object detection algorithms from images: A survey," *Array*, p. 100305, 2023.

[86] A. Shaw, D. Hunter, F. Landola, and S. Sidhu, "Squeezenas: Fast neural architecture search for faster semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.

[87] I. Salehin, M. S. Islam, P. Saha, S. Noman, A. Tuni, M. M. Hasan, and M. A. Baten, "Automl: A systematic review on automated machine learning with neural architecture search," *Journal of Information and Intelligence*, 2023.

[88] C. White, M. Safari, R. Sukthanker, B. Ru, T. Elsken, A. Zela, D. Dey, and F. Hutter, "Neural architecture search: Insights from 1000 papers," *arXiv preprint arXiv:2301.08727*, 2023.

[89] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, "Detnas: Backbone search for object detection," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[90] N. Wang, Y. Gao, H. Chen, P. Wang, Z. Tian, C. Shen, and Y. Zhang, "Nas-fcos: Fast neural architecture search for object detection," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 943–11 951.

[91] L. Chen, S. Lin, X. Lu, D. Cao, H. Wu, C. Guo, C. Liu, and F.-Y. Wang, "Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3234–3246, 2021.

[92] S. A. Nawaz, J. Li, U. A. Bhatti, M. U. Shoukat, and R. M. Ahmad, "Ai-based object detection latest trends in remote sensing, multimedia and agriculture applications," *Frontiers in Plant Science*, vol. 13, p. 1041514, 2022.

[93] J. Kang, S. Tariq, H. Oh, and S. S. Woo, "A survey of deep learning-based object detection methods and datasets for overhead imagery," *IEEE Access*, vol. 10, pp. 20 118–20 134, 2022.

[94] 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 Sensing*, vol. 13, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/1/89

[95] E. Arkin, N. Yadikar, X. Xu, A. Aysa, and K. Ubul, "A survey: Object detection methods from cnn to transformer," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21 353–21 383, 2023.

[96] H. Hakim and A. Fadhil, "Survey: convolution neural networks in object detection," in *Journal of Physics: Conference Series*, vol. 1804, no. 1. IOP Publishing, 2021, p. 012095.

[97] J. Ren and Y. Wang, "Overview of object detection algorithms using convolutional neural networks," *Journal of Computer and Communications*, vol. 10, no. 1, pp. 115–132, 2022.

[98] S. A. Singh, T. G. Meitei, and S. Majumder, "6 - short pcg classification based on deep learning," in *Deep Learning Techniques for Biomedical and Health Informatics*, B. Agarwal, V. E. Balas, L. C. Jain, R. C. Poonia, and Manisha, Eds. Academic Press, 2020, pp. 141–164. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128190616000069

[99] Q. Sellat, S. K. Bisoy, and R. Priyadarshini, "Chapter 10 - semantic segmentation for self-driving cars using deep learning: a survey," in *Cognitive Big Data Intelligence with a Metaheuristic Approach*, ser. Cognitive Data Science in Sustainable Computing, S. Mishra, H. K. Tripathy, P. K. Mallick, A. K. Sangaiah, and G.-S. Chae, Eds. Academic Press, 2022, pp. 211–238. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323851176000029

[100] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, "The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16 591–16 633, 2023.

[101] Y. Li, N. Miao, L. Ma, F. Shuang, and X. Huang, "Transformer for object detection: Review and benchmark," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107021, 2023.

[102] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[103] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022.

[104] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[105] R. Halfvordsson, J. Nordh, A. Suhren Gustafsson, J. Wall, M. Westerberg, and A. Wirehed, "Generative adversarial networks for object detection in ad/adas functions," 2019.

[106] S. M. Al Jaberi, A. Patel, and A. N. AL-Masri, "Object tracking and detection techniques under gann threats: A systemic review," *Applied Soft Computing*, p. 110224, 2023.

[107] H. Wu, H. Zhao, and M. Zhang, "Not all attention is all you need," *arXiv preprint arXiv:2104.04692*, 2021.

[108] J. Beal, E. Kim, E. Tzeng, D. H. Park, A. Zhai, and D. Kislyuk, "Toward transformer-based object detection," *arXiv preprint arXiv:2012.09958*, 2020.

[109] A. Chiatti, E. Motta, E. Daga, and G. Bardaro, "Fit to measure: Reasoning about sizes for robust object recognition," *arXiv preprint arXiv:2010.14296*, 2020.

[110] X. Luo, H. Li, and S. Lee, "Bridging the gap: Neuro-symbolic computing for advanced ai applications in construction," *Frontiers of Engineering Management*, pp. 1–9, 2023.

[111] M. Garnelo and M. Shanahan, "Reconciling deep learning with symbolic artificial intelligence: representing objects and relations," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 17–23, 2019, artificial Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352154618301943

[112] S. Badreddine, A. d. Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, p. 103649, 2022.

[113] F. Manigrasso, F. D. Miro, L. Morra, and F. Lamberti, "Faster-ltn: a neuro-symbolic, end-to-end object detection architecture," in *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part II 30.* Springer, 2021, pp. 40–52.

[114] P. Zamboni, J. M. Junior, J. d. A. Silva, G. T. Miyoshi, E. T. Matsubara, K. Nogueira, and W. N. Gonçalves, "Benchmarking anchor-based and anchor-free state-of-the-art deep learning methods for individual tree detection in rgb high-resolution images," *Remote Sensing*, vol. 13, no. 13, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/13/2482

[115] A. B. Amjoud and M. Amrouch, "Object detection using deep learning, cnns and vision transformers: A review," *IEEE Access*, 2023.

[116] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[117] W. Liang, P. Xu, L. Guo, H. Bai, Y. Zhou, and F. Chen, "A survey of 3d object detection," *Multimedia Tools and Applications*, vol. 80, no. 19, pp. 29 617–29 641, 2021.

[118] H.-L. Lee, Y.-j. Kim, B.-G. Kim *et al.*, "A survey for 3d object detection algorithms from images," *Journal of Multimedia Information System*, vol. 9, no. 3, pp. 183–190, 2022.

[119] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[120] T. Wu and Y. Dong, "Yolo-se: Improved yolov8 for remote sensing object detection and recognition," *Applied Sciences*, vol. 13, no. 24, p. 12977, 2023.

[121] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," in *Computer Vision and Pattern Recognition.* Springer Berlin/Heidelberg, Germany, 2018, pp. 1804–02 767.

[122] L. Zhao and S. Li, "Object detection algorithm based on improved yolov3," *Electronics*, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/537

[123] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14.* Springer, 2016, pp. 21–37.

[124] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[125] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[126] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[127] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[128] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[129] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[130] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[131] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of detr with spatially modulated co-attention," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3621–3630.

[132] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor detr: Query design for transformer-based object detection," *arXiv preprint arXiv:2109.07107*, vol. 3, no. 6, 2021.

[133] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 183–26 197, 2021.

[134] T. Shehzadi, K. A. Hashmi, D. Stricker, and M. Z. Afzal, "2d object detection with transformers: A review," *arXiv preprint arXiv:2306.04670*, 2023.

[135] M. Iman, H. R. Arabnia, and K. Rasheed, "A review of deep transfer learning and recent advancements," *Technologies*, vol. 11, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2227-7080/11/2/40

[136] A. Balasubramaniam and S. Pasricha, "Object detection in autonomous vehicles: Status and open challenges," *arXiv preprint arXiv:2201.07706*, 2022.

[137] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[138] Y. Wang, Z. Liu, and S. Lian, "Semi-supervised object detection: A survey on recent research and progress," *arXiv preprint arXiv:2306.14106*, 2023.

[139] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," *arXiv preprint arXiv:2006.05278*, 2020.

[140] N. Le, V. S. Rathour, K. Yamazaki, K. Luu, and M. Savvides, "Deep reinforcement learning in computer vision: a comprehensive survey," *Artificial Intelligence Review*, pp. 1–87, 2022.

[141] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[142] X. Liu, H. Yang, A. Ravichandran, R. Bhotika, and S. Soatto, "Multi-task incremental learning for object detection," 2020.

[143] X. Zhao, S. Schulter, G. Sharma, Y.-H. Tsai, M. Chandraker, and Y. Wu, "Object detection with a unified label space from multiple datasets," 2020.

[144] Y. Chen, M. Wang, A. Mittal, Z. Xu, P. Favaro, J. Tighe, and D. Modolo, "Scaledet: A scalable multi-dataset object detector," 2023.

[145] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[146] A. Alkhulaifi, F. Alsahli, and I. Ahmad, “Knowledge distillation in deep learning and its applications,” *PeerJ Computer Science*, vol. 7, p. e474, 2021.

[147] P. Oza, V. A. Sindagi, V. V. Sharmini, and V. M. Patel, “Unsupervised domain adaptation of object detectors: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[148] Y. Zhou, S. Wen, D. Wang, J. Mu, and I. Richard, “Object detection in autonomous driving scenarios based on an improved faster-rcnn,” *Applied Sciences*, vol. 11, no. 24, p. 11630, 2021.

[149] N. Arora and Y. Kumar, “Automatic vehicle detection system in day and night mode: challenges, applications and panoramic review,” *Evolutionary Intelligence*, vol. 16, no. 4, pp. 1077–1095, 2023.

[150] S. Campbell, N. O’Mahony, L. Krpalcova, D. Riordan, J. Walsh, A. Murphy, and C. Ryan, “Sensor technology in autonomous vehicles: A review,” in *2018 29th Irish Signals and Systems Conference (ISSC)*. IEEE, 2018, pp. 1–4.

[151] M. Karg and C. Scharfenberger, “Deep learning-based pedestrian detection for automated driving: achievements and future challenges,” in *Development and Analysis of Deep Learning Architectures*. Springer, 2019, pp. 117–143.

[152] D.-T. Iancu, A. Sorici, and A. M. Florea, “Object detection in autonomous driving - from large to small datasets,” in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2019, pp. 1–6.

[153] X. Wu, D. Sahoo, and S. C. Hoi, “Recent advances in deep learning for object detection,” *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[154] M. Liu, E. Yurtsever, X. Zhou, J. Fossaert, Y. Cui, B. L. Zagar, and A. C. Knoll, “A survey on autonomous driving datasets: Data statistic, annotation, and outlook,” *arXiv preprint arXiv:2401.01454*, 2024.

[155] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” 2020.

[156] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE transactions on pattern analysis and machine intelligence.*, vol. 34, no. 4, pp. 743–761, 2011.

[157] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[158] S. Zhang, R. Benenson, and B. Schiele, “Citypersons: A diverse dataset for pedestrian detection,” 2017.

[159] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.

[160] R. Qian, X. Lai, and X. Li, “3d object detection for autonomous driving: A survey,” *Pattern Recognition*, vol. 130, p. 108796, 2022.

[161] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, “The apolloscape open dataset for autonomous driving and its application,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, p. 2702–2719, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2019.2926463

[162] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[163] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” 2020.

[164] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, “Scalability in perception for autonomous driving: Waymo open dataset,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2443–2451.

[165] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, “A review and comparative study on probabilistic object detection in autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9961–9980, 2021.

[166] Y. Zhang, X. Song, B. Bai, T. Xing, C. Liu, X. Gao, Z. Wang, Y. Wen, H. Liao, G. Zhang *et al.*, “2nd place solution for waymo open dataset challenge–real-time 2d object detection,” *arXiv preprint arXiv:2106.08713*, 2021.

[167] X. Han, J. Chang, and K. Wang, “Real-time object detection based on yolo-v2 for tiny vehicle object,” *Procedia Computer Science*, vol. 183, pp. 61–72, 2021.

[168] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, “Transfusion: Robust lidar-camera fusion for 3d object detection with transformers,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1090–1099.

[169] X. Feng, H. Du, H. Fan, Y. Duan, and Y. Liu, “Seformer: Structure embedding transformer for 3d object detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 632–640.

[170] A. Gupta, K. Illanko, and X. Fernando, “Object detection for connected and autonomous vehicles using cnn with attention mechanism,” in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–6.

[171] T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using yolo: Challenges, architectural successors, datasets and applications,” *multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023.

[172] S. Zhou, M. Xie, Y. Jin, F. Miao, and C. Ding, “An end-to-end multi-task object detection using embedded gpu in autonomous driving,” in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 122–128.

[173] B. B. Elallid, N. Benamar, A. S. Hafid, T. Rachidi, and N. Mrani, “A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 7366–7390, 2022.

[174] J. Mao, S. Shi, X. Wang, and H. Li, “3d object detection for autonomous driving: A comprehensive survey,” *International Journal of Computer Vision*, pp. 1–55, 2023.

[175] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” *IEEE access*, vol. 7, pp. 128 837–128 868, 2019.

[176] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: A review,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3388–3415, 2020.

[177] D. Bogdoll, S. Uhlemeyer, K. Kowol, and J. M. Zöllner, “Perception datasets for anomaly detection in autonomous driving: A survey,” *arXiv preprint arXiv:2302.02790*, 2023.

[178] A. Tharwat and W. Schenck, “A survey on active learning: State-of-the-art, practical challenges and research directions,” *Mathematics*, vol. 11, no. 4, p. 820, 2023.

[179] D. U. Jo, S. Yun, and J. Y. Choi, “How much a model be trained by passive learning before active learning?” *IEEE Access*, vol. 10, pp. 34 677–34 689, 2022.

[180] M. Wu, C. Li, and Z. Yao, “Deep active learning for computer vision tasks: Methodologies, applications, and challenges,” *Applied Sciences*, vol. 12, no. 16, p. 8103, 2022.

[181] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and S. Y. Philip, “Active learning: A survey,” in *Data classification*. Chapman and Hall/CRC, 2014, pp. 599–634.

[182] A. Matosevic, “Batch active learning for deep object detection in videos,” 2021.

[183] D. Garcia, J. Carias, T. Adão, R. Jesus, A. Cunha, and L. G. Magalhães, “Ten years of active learning techniques and object detection: A systematic review,” *Applied Sciences*, vol. 13, no. 19, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/19/10667

[184] J. Flabeau, “Deep active learning of object detection for smart city,” 2020.

[185] R. Takezoe, X. Liu, S. Mao, M. T. Chen, Z. Feng, S. Zhang, X. Wang *et al.*, “Deep active learning for computer vision: Past and future,” *APSIPA Transactions on Signal and Information Processing*, vol. 12, no. 1, 2023.

[186] T. Wan, K. Xu, T. Yu, X. Wang, D. Feng, B. Ding, and H. Wang, “A survey of deep active learning for foundation models,” *Intelligent Computing*, vol. 2, 11 2023.

[187] S. Mohamadi and H. Amindavar, “Deep bayesian active learning, a brief survey on recent advances,” *arXiv preprint arXiv:2012.08044*, 2020.

[188] Z. Qu, J. Du, Y. Cao, Q. Guan, and P. Zhao, “Deep active learning for remote sensing object detection,” *arXiv: Computer Vision and Pattern Recognition*, 2020.

[189] T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji, and Q. Ye, “Multiple instance active learning for object detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5326–5335.

[190] A. L. Chandra, S. V. Desai, V. N. Balasubramanian, S. Ninomiya, and W. Guo, “Active learning with point supervision for cost-effective panicle detection in cereal crops,” *Plant Methods.*, vol. 16, no. 1, pp. 1–16, 2020.

[191] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, “Deep active learning for efficient training of a lidar 3d object detector,” in *2019 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2019, pp. 667–674.

[192] S. Roy, A. Unmesh, and V. P. Namboodiri, “Deep active learning for object detection,” in *BMVC*, vol. 362. BMVA, 2018, p. 91.

[193] A. Kirsch, J. v. Amersfoort, and Y. Gal, *BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning.* Red Hook, NY, USA: Curran Associates Inc., 2019.

[194] S. H. Park and S. B. Kim, “Robust expected model change for active learning in regression,” *Applied Intelligence*, vol. 50, no. 2, pp. 296–313, 2020.

[195] Y. Zhao, Z. Shi, J. Zhang, D. Chen, and L. Gu, “A novel active learning framework for classification: using weighted rank aggregation to achieve multiple query criteria,” *Pattern Recognition*, vol. 93, pp. 581–602, 2019.

[196] Y. Li, B. Fan, W. Zhang, W. Ding, and J. Yin, “Deep active learning for object detection,” *Information Sciences*, vol. 579, pp. 418–433, 2021.

[197] X. Gui, X. Lu, and G. Yu, “Cost-effective batch-mode multi-label active learning,” *Neurocomputing*, vol. 463, pp. 355–367, 2021.

[198] J. Wu, V. S. Sheng, J. Zhang, H. Li, T. Dadakova, C. L. Swisher, Z. Cui, and P. Zhao, “Multi-label active learning algorithms for image classification: Overview and future promise,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–35, 2020.

[199] T. Yao, W. Wang, and Y. Gu, “A deep multiview active learning for large-scale image classification,” *Mathematical Problems in Engineering*, vol. 2020, 2020.

[200] C. Su, Z. Yan, and G. Yu, “Cost-effective multi-instance multilabel active learning,” *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7177–7203, 2021.

[201] G. Yu, Y. Xing, J. Wang, C. Domeniconi, and X. Zhang, “Multiview multi-instance multilabel active learning,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.

[202] J. Guo, Z. Pang, M. Bai, P. Xie, and Y. Chen, “Dual generative adversarial active learning,” *Applied Intelligence*, pp. 1–12, 2021.

[203] J. Wang, Y. Yan, Y. Zhang, G. Cao, M. Yang, and M. K. Ng, “Deep reinforcement active learning for medical image classification,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2020, pp. 33–42.

[204] N. Nadagouda, A. Xu, and M. A. Davenport, “Active metric learning and classification using similarity queries,” *arXiv preprint arXiv:2202.01953*, 2022.

[205] N. Zemmal, N. Azizi, M. Sellami, S. Cheriguene, A. Ziani, M. AlDwairi, and N. Dendani, “Particle swarm optimization based swarm intelligence for active learning improvement: Application on medical data classification,” *Cognitive Computation*, vol. 12, no. 5, pp. 991–1010, 2020.

[206] Z. Deng, Y. Yang, K. Suzuki, and Z. Jin, “Fedal: An federated active learning framework for efficient labeling in skin lesion analysis,” in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 1554–1559.

[207] Q. Jin, M. Yuan, Q. Qiao, and Z. Song, “One-shot active learning for image segmentation via contrastive learning and diversity-based sampling,” *Knowledge-Based Systems*, vol. 241, p. 108278, 2022.

[208] N. Bougie and R. Ichise, “Goal-driven active learning,” in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS ’22. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2022, p. 1923–1925.

[209] J. Shim and S. Kang, “Domain-adaptive active learning for cost-effective virtual metrology modeling,” *Computers in Industry*, vol. 135, p. 103572, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361521001792

[210] J. Bommel, “Active learning during federated learning for object detection,” B.S. thesis, University of Twente, 2021.

[211] K. Wang, L. Lin, X. Yan, Z. Chen, D. Zhang, and L. Zhang, “Cost-effective object detection: Active sample mining with switchable selection criteria,” *IEEE transactions on neural networks and learning systems.*, vol. 30, no. 3, pp. 834–850, 2018.

[212] P. Mi, J. Lin, Y. Zhou, Y. Shen, G. Luo, X. Sun, L. Cao, R. Fu, Q. Xu, and R. Ji, “Active teacher for semi-supervised object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 482–14 491.

[213] H. V. Vo, O. Siméoni, S. Gidaris, A. Bursuc, P. Pérez, and J. Ponce, “Active learning strategies for weakly-supervised object detection,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*. Springer, 2022, pp. 211–230.

[214] J. K. Mandivarapu, B. Camp, and R. Estrada, “Deep active learning via open-set recognition,” *Frontiers in Artificial Intelligence*, vol. 5, p. 2, 2022.

[215] K.-P. Ning, X. Zhao, Y. Li, and S.-J. Huang, “Active learning for open-set annotation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 41–49.

[216] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, “A survey of deep active learning,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[217] P. Kumar and A. Gupta, “Active learning query strategies for classification, regression, and clustering: a survey,” *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 913–945, 2020.

[218] W. Yu, S. Zhu, T. Yang, and C. Chen, “Consistency-based active learning for object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3951–3960.

[219] J. Wu, J. Chen, and D. Huang, “Entropy-based active learning for object detection with progressive diversity constraint,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9397–9406.

[220] S. Sivaraman and M. M. Trivedi, “Active learning for on-road vehicle detection: A comparative study,” *Machine vision and applications.*, vol. 25, no. 3, pp. 599–611, 2014.

[221] T. Yang, J. Li, Q. Pan, C. Zhao, and Y. Zhu, “Active learning based pedestrian detection in real scenes,” in *18th International Conference on Pattern Recognition (ICPR’06)*, vol. 4, 2006, pp. 904–907.

[222] A. Moses, S. Jakkampudi, C. Danner, and D. Biega, "Localization-based active learning (local) for object detection in 3d point clouds," in *Geospatial Informatics XII*, vol. 12099. SPIE, 2022, pp. 44–58.

[223] Z. Liang, X. Xu, S. Deng, L. Cai, T. Jiang, and K. Jia, "Exploring diversity-based active learning for 3d object detection in autonomous driving," *arXiv preprint arXiv:2205.07708*, 2022.

[224] D. L. Li, M. Prasad, C.-L. Liu, and C.-T. Lin, "Multi-view vehicle detection based on fusion part model with active learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3146–3157, 2021.

[225] U. Aggarwal, "Active and incremental deep learning with class imbalanced data," Ph.D. dissertation, Université Paris-Saclay, 2022.

[226] C.-A. Brust, C. Käding, and J. Denzler, "Active and incremental learning with weak supervision," *KI-Künstliche Intelligenz.*, pp. 1–16, 2020.

[227] T. N. Cardoso, R. M. Silva, S. Canuto, M. M. Moro, and M. A. Gonçalves, "Ranked batch-mode active learning," *Information Sciences.*, vol. 379, pp. 313–337, 2017.

[228] Y. Gu, D. Zydek, and Z. Jin, "Active learning based on random forest and its application to terrain classification," in *Progress in Systems Engineering*, ser. Advances in Intelligent Systems and Computing, H. Selvaraj, D. Zydek, and G. Chmaj, Eds. Springer, 2015, vol. 366, pp. 273–278.

[229] W. Kuo, C. Häne, E. Yuh, P. Mukherjee, and J. Malik, "Cost-sensitive active learning for intracranial hemorrhage detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2018, pp. 715–723.

[230] C. Sachdeva and S. Goel, "An improved approach for solving 0/1 knapsack problem in polynomial time using genetic algorithms," in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, 2014, pp. 1–4.

[231] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[232] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations*, 2018.

[233] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE.*, vol. 109, no. 1, pp. 43–76, 2020.

[234] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II 13.* Springer, 2015, pp. 613–627.

[235] N. Ragesh and R. Rajesh, "Pedestrian detection in automotive safety: Understanding state-of-the-art," *IEEE Access*, vol. 7, pp. 47 864–47 890, 2019.

[236] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3626–3633.

[237] W. Nam, B. Han, and J. H. Han, "Improving object localization using macrofeature layout selection," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1801–1808.

[238] T. Song, L. Sun, D. Xie, H. Sun, and S. Pu, "Small-scale pedestrian detection based on somatic topology localization and temporal feature aggregation," *arXiv preprint arXiv:1807.01438*, 2018.