

Università di Genova

PhD Program in Science and Technology for Electronic and Telecommunication Engineering

New Approaches of eXplainable AI: From Video Analytics to Federated Learning

Ahmad Samandari

Thesis submitted for the degree of *Doctor of Philosophy* (38° cycle)

26 Feb 2026

Prof. Fabio Patrone	Tutor
Dott. Maurizio Mongelli	Co-tutor
Prof. Maurizio Valle	Head of the PhD program









Borsa di dottorato cofinanziata con risorse dell'Unione europea – NextGenerationEU
Piano Nazionale di Ripresa e Resilienza, Missione 4, componente 2
("Dalla Ricerca all'Impresa")



Università di Genova



Consiglio Nazionale delle Ricerche

artificial intelligence by aitek

Department of Electrical, Electronics and Telecommunications Engineering and Naval Architecture, University of Genoa

Institute of Electronics, Computer and Telecommunication Engineering, Italian National Research Council

Aitek S.p.A.

Nuovi approcci di intelligenza artificiale spiegabile: dalla video-analisi al Federated Learning

Ahmad Samandari

Relatore

Prof. Fabio Patrone (Università di Genova)

Correlatore

Dott. Maurizio Mongelli (Consiglio Nazionale delle Ricerche)

Relatore industriale

Dott. Stefano Delucchi (Aitek S.p.A.)

New Approaches of eXplainable AI: From Video Analytics to Federated Learning

Ahmad Samandari

⋈

Committee:

President of the committee
Fabio Patrone , Assistant Professor, University of Genoa, Italy

Reviewers
Tomaso de Cola, Team Leader, Project Manager, German Aerospace Center (DLR), Germany
Luigi Pomante, Assistant Professor, Università degli Studi dell'Aquila, Italy

Tutors
Fabio Patrone, Assistant Professor, University of Genoa, Italy
Maurizio Mongelli, Senior Researcher, Consiglio Nazionale delle Ricerche (CNR), Italy

Stefano Delucchi, Industrial Advisor, Aitek S.p.A., Italy

University of Genoa

Ahmad Samandari

New Approaches of eXplainable AI: From Video Analytics to Federated Learning

xxiii+182 p.

To my family, with love and gratitude.

Abstract

This thesis investigates how eXplainable Artificial Intelligence (XAI) can be systematically embedded into Machine Learning (ML) pipelines for safety–critical applications. The central argument is that explainability should not be treated as a post-hoc feature but as a design principle that governs how systems are constructed, monitored, and deployed. The work spans two domains: video analytics for autonomous and assistive mobility, and distributed learning under privacy and ownership constraints.

On the video analytics side, strong baselines were established by fine-tuning YOLOv8s for indoor mobility, achieving high accuracy in detecting people and wheelchairs. Building on this backbone, an Operational Design Domain (ODD) Checker was introduced, combining visual feature analysis with Decision Tree (DT) rules to provide interpretable, auditable safety monitoring. Explainability was extended to scene-level reasoning through benchmarking of Vision–Language Models (CLIP, MiniGPT-4, GPT-4V), which were evaluated on their ability to classify navigation scenes as "Safe to Proceed" or "Risky to Proceed" and contextualized within the swan metaphor for rare risks. To quantify predictive reliability, Conformal Prediction (CP) was applied to object detection, establishing finite-sample coverage guarantees and demonstrating the trade-offs between box-wise and image-wise Uncertainty Quantification (UQ) strategies.

In distributed learning, the thesis introduces Federated Learning with Interpretable Rule Transfer (FL-IRT), a framework that replaces opaque parameter averaging with the construction of rule-based models at client and server levels. FL-IRT enables global models that are both competitive in accuracy and transparent in logic, while supporting secure aggregation and GDPR-compliant privacy mechanisms. Experimental results across multiple datasets confirm its scalability, robustness to non-iid conditions, and significant efficiency gains over neural baselines.

Taken together, these contributions show that explainability can be embedded across abstraction layers—from pixel-level features and object detection to semantic reasoning, statistical calibration, and distributed learning. By advancing interpretable video analytics, principled UQ, and transparent federated frameworks, the thesis demonstrates that trustworthy-by-design AI is achievable without prohibitive sacrifices in accuracy or efficiency. The broader implication is that XAI functions as a regulatory layer in AI, transforming abstract principles of accountability and safety into enforceable engineering standards.

Keywords: eXplainable Artificial Intelligence; Video Analytics; Perception Monitoring; Operational Design Domain; Vision–Language Models; Conformal Prediction; Federated Learning; Decision Tree; Interpretable Rule Transfer; Safety–Critical AI

Abstract

Questa tesi indaga come l'eXplainable Artificial Intelligence (XAI) possa essere sistematicamente incorporata nelle pipeline di Machine Learning (ML) per applicazioni in scenari safety–critical. L'argomento centrale è che l'explainability non debba essere trattata come una funzionalità post-hoc, ma come un principio di progettazione che guida la costruzione, il monitoraggio e il dispiegamento dei sistemi. Il lavoro si sviluppa in due domini: la video analytics per la mobilità autonoma e assistiva, e l'apprendimento distribuito in presenza di vincoli di privacy e proprietà dei dati.

Nel contesto della video analytics, sono state definite solide basi addestrando e ottimizzando YOLOv8s per ambienti indoor, con elevate prestazioni nel rilevamento di persone e sedie a rotelle. Su questo backbone è stato sviluppato un Operational Design Domain (ODD) Checker, che combina l'analisi delle feature visive con regole basate su alberi di decisione (DT) per un monitoraggio interpretabile e verificabile della sicurezza. L'explainability è stata poi estesa al ragionamento a livello di scena tramite una valutazione comparativa di modelli Vision–Language (CLIP, MiniGPT-4, GPT-4V), capaci di classificare ambienti di navigazione come "Safe to Proceed" o "Risky to Proceed" e contestualizzati nel paradigma dei cigni per la gestione dei rischi rari. Per quantificare l'affidabilità predittiva, la Conformal Prediction (CP) è stata applicata al rilevamento di oggetti, fornendo garanzie statistiche finite e mettendo in luce i compromessi tra strategie di quantificazione dell'incertezza (Uncertainty Quantification, UQ) box-wise e image-wise.

Nel dominio dell'apprendimento distribuito, la tesi introduce Federated Learning with Interpretable Rule Transfer (FL-IRT), un framework che sostituisce l'aggregazione opaca dei parametri con la costruzione di modelli basati su regole sia lato client che lato server. FL-IRT consente di ottenere modelli globali competitivi in termini di accuratezza ma anche trasparenti nel processo decisionale, supportando al contempo meccanismi di aggregazione sicura e conformità con la normativa GDPR. Esperimenti condotti su diversi dataset confermano la sua scalabilità, la robustezza in condizioni non-iid e un notevole miglioramento di efficienza rispetto ai baseline neurali.

Complessivamente, questi contributi dimostrano che l'explainability può essere integrata a diversi livelli di astrazione—dalle feature pixel–level e dal rilevamento di oggetti, fino al ragionamento semantico, alla calibrazione statistica e all'apprendimento distribuito. Avanzando la video analytics interpretabile, la quantificazione dell'incertezza basata su principi formali e i framework federati trasparenti, la tesi mostra che l'AI trustworthy-by-design è realizzabile senza sacrifici proibitivi in termini di accuratezza o efficienza. L'implicazione più ampia è che la XAI funge da livello regolatorio nell'AI, trasformando principi astratti di responsabilità e sicurezza in standard ingegneristici applicabili.

Parole chiave: eXplainable Artificial Intelligence; Video Analytics; Monitoraggio della Percezione; Operational Design Domain; Vision–Language Models; Conformal Prediction; Federated Learning; Decision Tree; Interpretable Rule Transfer; AI per Sistemi Safety–Critical

Contents

List of Figures xv

List of Tables xix

Notations xxi

Acronyms xxiii

1 Introduction 1
- 1.1 eXplainable Artificial Intelligence (XAI) . . . 1
- 1.2 Video Analytics in Automated Driving Systems . . . 3
- 1.3 Federated Learning (FL) . . . 5
- 1.4 Contributions of this Thesis . . . 8
- 1.5 Structure of the Thesis . . . 12

2 Previous work 13
- 2.1 Perception and Vision in Autonomous Driving Systems (ADS) . . . 13
- 2.2 Extending and Monitoring the Operational Design Domain in Autonomous Driving Systems . . . 15
- 2.3 Scene Comprehension and Safety Monitoring with Vision-Language Model (VLM) . . . 16
- 2.4 Uncertainty Quantification through Conformal Prediction . . . 18
- 2.5 Explainability in Federated Learning Systems . . . 19
- 2.6 Summary and Research Gaps . . . 22

3 Perception and Vision in Autonomous Driving Systems 25
- 3.1 Introduction . . . 25
  - 3.1.1 Background . . . 25
  - 3.1.2 Open Challenges . . . 26
  - 3.1.3 Contributions . . . 26
- 3.2 Methodology . . . 27
  - 3.2.1 Evolution of YOLO Models . . . 27
  - 3.2.2 YOLOv8 Architecture . . . 28
  - 3.2.3 YOLOv8 Model Variants . . . 28
  - 3.2.4 Fine-Tuning for Domain Adaptation . . . 29
- 3.3 Experimental Validation . . . 30
  - 3.3.1 Datasets . . . 30
  - 3.3.2 Training Configuration . . . 35

3.3.3 Evaluation Metrics and Performance Curves 35
3.3.4 Results 37
3.4 Conclusions 45

4 eXplainable Checker of Video Analytics Performance in Indoor Smart Mobility 47
4.1 Introduction 47
4.1.1 Background 47
4.1.2 Key Issues in Autonomous Driving Systems (ADS) Safety 48
4.1.3 Open Challenges 48
4.1.4 Explainable Artificial Intelligence (XAI) for Safe Autonomous Driving Systems (ADS) 49
4.1.5 Contributions 50
4.2 Methodology 50
4.2.1 Object Detection 51
4.2.2 Feature Extraction 52
4.2.3 Validation 52
4.2.4 Feature Selection and Rule Extraction 54
4.2.5 Real-Time Certainty Evaluation 55
4.3 Experimental validation 55
4.3.1 Object Detector: Dataset and Configuration 55
4.3.2 ODD Checker Dataset 55
4.3.3 Results 56
4.4 Conclusions 68

5 Vision-Language Models for Scene Comprehension Support in Autonomous Wheelchair Navigation 71
5.1 Introduction 71
5.1.1 Background and Motivation 71
5.1.2 Vision-Language Model (VLM): Principles and Architecture . . . 72
5.1.3 Rare and Unforeseen Risks: The Swan Metaphor 72
5.1.4 Contributions of This Chapter 74
5.2 Methodology 75
5.2.1 Model Architectures and Reasoning Pipelines 75
5.2.2 Prompt Engineering and Configuration Tuning 78
5.2.3 Implementation Details 80
5.3 Experimental validation 81
5.3.1 Dataset 81
5.3.2 Results 82
5.4 Conclusion 85

6 Uncertainty Quantification through Conformal Prediction 89
6.1 Introduction . . . 89
6.1.1 Background and Motivation . . . 89
6.1.2 Conformal Prediction . . . 90
6.1.3 Contributions . . . 92
6.2 Methodology . . . 93
6.2.1 Conformal Prediction Fundamentals . . . 93
6.2.2 Box-wise Conformal Prediction . . . 94
6.2.3 Image-wise Conformal Prediction and Conformal Risk Control . . . 98
6.3 Experimental Validation . . . 100
6.3.1 Datasets . . . 101
6.3.2 Evaluation Metrics . . . 101
6.3.3 Results . . . 102
6.4 Conclusion . . . 110

7 Explainability in Federated Learning Systems 113
7.1 Introduction . . . 113
7.1.1 Scope and Motivation . . . 113
7.1.2 Contributions . . . 114
7.2 Methodology . . . 115
7.2.1 Definitions . . . 116
7.2.2 Client-side Algorithm . . . 118
7.2.3 Server-side Algorithm . . . 120
7.3 Experimental validation . . . 123
7.3.1 Dataset . . . 123
7.3.2 Dataset Statistical Properties . . . 127
7.3.3 Baseline and Comparative Models . . . 127
7.3.4 Hardware and Software Setup . . . 128
7.3.5 Simulation Settings and Model Configurations . . . 128
7.3.6 Results . . . 130
7.4 Conclusion . . . 144

8 General conclusion 147

Appendix

155

Bibliography 159

List of Figures

1.1 Typical Federated Learning (FL) workflow. Clients train local models on their private data and send only parameter updates ($W_1, W_2, \ldots, W_n$) to a central server, which aggregates them (e.g., using Federated Averaging (FedAvg)) into a global model W'. The updated global model is then redistributed back to clients. 6
1.2 Main categories of FL according to data distribution. (a) Horizontal FL: clients share the same feature space but different samples. (b) Vertical FL: clients share the same samples but different feature sets. (c) Federated Transfer Learning: little overlap in both features and samples, requiring transfer learning techniques. 7
1.3 Deployment scenario in the REliable and eXplainable Swarm Intelligence for People with Reduced mObility (REXASI-PRO) framework: collaboration between an autonomous wheelchair and a drone for safe indoor navigation. 9
1.4 Risk-aware perception in assistive mobility: vision-based analytics detect safety-critical situations (e.g., occluded hazards, crowding) and support runtime monitoring for trustworthy decision making within REXASI-PRO. 10

3.1 Model Structure of Yolov8 29
3.2 Examples from Open Images Subset – Wheelchair 31
3.3 Examples from Open Images Subsets – Bicycle, Chair, and Desk 32
3.4 Example from Roboflow Wheelchair Dataset 33
3.5 Examples from Penn-Fudan Pedestrian Dataset 34
3.6 Examples from the CNR Villa Dataset 34
3.7 Intersection over Union (IoU) 36
3.8 Confusion matrix (normalized) 39
3.9 F1-confidence curve 40
3.10 Precision-confidence curve 40
3.11 Recall-confidence curve 41
3.12 Precision–Recall curve 41
3.13 Confusion matrix (CNR Villa dataset) 43
3.14 F1-confidence curve (CNR Villa) 44
3.15 Precision–Recall curve (CNR Villa) 44
3.16 Qualitative detection results on the CNR Villa dataset 45

4.1 Real-time ODD Monitoring System Architecture 51
4.2 Example perturbations 57
4.3 Example from Aitek SpA dataset 58

4.4 Trade-off analysis of False Negative Rate (FNR) and False Positive Rate (FPR) for different feature set sizes in the Person and Wheelchair categories. Optimal performance is achieved at 20 features. 59
4.5 Top 20 most informative features for the Person category based on cumulative ranking after 100 CFR iterations. 59
4.6 Top 20 most informative features for the Wheelchair category based on cumulative ranking after 100 CFR iterations. 60
4.7 Representative frames from the two video sequences. Top row: Video 1 under nominal indoor conditions. Bottom row: Video 2 under synthetic perturbations (e.g., extreme brightness shifts). 66
4.8 Video 1: Checker predictions vs. ground truth under nominal conditions. 67
4.9 Video 2: Checker predictions vs. ground truth under perturbed conditions. 68

5.1 Conceptual architecture of a Vision-Language Model (VLM). Image inputs and text prompts are encoded into a shared representation space, where they are jointly processed for multimodal reasoning [1]. 73
5.2 Swans in risk analysis. Grey swans refer to rare but partially understood risk conditions. Black swans represent fundamentally unknowable hazards that may remain invisible to structured evaluation frameworks. 74
5.3 Model architectures and reasoning pipelines. (a) Modular Contrastive Language-Image Pre-Training (CLIP) with rule-based fusion of safety cues. (b) Decision-making using MiniGPT-4 and GPT-4 with Vision (GPT-4V). (c) Traditional two-stage perception pipeline. 76
5.4 Representative dataset examples: (a) risky scenes with obstacles or high crowd density; (b) safe scenes with clear forward paths; (c) safe scenes from KTH-IDOL2; (d) perturbed scenes with fog or smoke, labeled as risky due to impaired visibility. 81
5.5 Scene labeled as *Risky to Proceed* by MiniGPT-4 and CLIP; CLIP's rationale was incorrect. GPT-4V misclassified the scene as *Safe to Proceed*. 83
5.6 Borderline scene with no immediate obstacles or crowd, misclassified by all models. 84
5.7 Ambiguous scenario: a person in a wheelchair near the forward path introduces uncertainty in an otherwise semi-open environment. 84

6.1 Conformal Prediction (CP) in image classification: prediction sets shrink for easy inputs and expand for ambiguous ones. 91
6.2 Conformal Prediction (CP) intervals for object detection. Red boxes denote ground-truth annotations, while shaded regions show conformalized bounding boxes with probabilistic coverage guarantees [2]. 92
6.3 Box-wise conformalization illustrated through Additive and Multiplicative nonconformity scores [3]. 96

6.4 Comparison of Bonferroni correction and max-residual conformalization. Bonferroni estimates four coordinate-wise quantiles at level $\alpha/4$, leading to conservative margins, whereas the max-residual method uses a single quantile at level α, yielding tighter bounds [3]. 97
6.5 Box-wise examples at $\alpha = 0.1$ 103
6.6 Coverage vs. α for box-wise methods (Test set) 105
6.7 Stretch vs. α for box-wise methods (Test set) 106
6.8 Representative effect of α on box-wise conformalization (Test set) 107
6.9 Image-wise examples at $\alpha = 0.3$ (Test set) 109

7.1 Overview of the Federated Learning for Interpretable Rule Trees (FL-IRT) framework. 116
7.2 3D Scatterplot of the Hospital dataset (HSP) dataset: homogeneous distribution in the two features X_1, X_2 with non-Independent and Identically Distributed (non-IID) patterns in the *Age* feature. The Parameter Server (PS) has to learn the clusters shift, depending on *Age*. 125
7.3 Examples of attribute relevance rankings for collision (i.e., collision label equal to 1) are depicted for Group 1 (left) and Group 2 (right), concerning the *Vehicle Platooning* dataset. Different rankings mean different rule sets, typically motivated by different statistical behaviors of data (between the two groups). 126
7.4 HSP dataset: test metrics across all communication rounds for the 24-client configuration. 132
7.5 Vehicle Platooning dataset (VP) dataset: test metrics across all communication rounds for the 24-client configuration under statistical heterogeneity. 133
7.6 Modified National Institute of Standards and Technology dataset (MNIST) dataset: test metrics across all communication rounds for the 24-client configuration. 135
7.7 Adult dataset (ADL) dataset: test metrics across all communication rounds for the 4-client configuration. 136
7.8 Average FPR (top left), FNR (top right), Accuracy (ACC) (bottom left), and misclassification rate (bottom right) of the Decision Tree (DT) method for an increasing number of samples of the training VP dataset. 140

List of Tables

3.1 Performance of YOLOv8 variants . . . 30
3.2 Validation performance of fine-tuned YOLOv8s . . . 38
3.3 Performance on the CNR Villa dataset . . . 42

4.1 Validation dataset breakdown . . . 56
4.2 Person Feature Statistics . . . 61
4.3 Wheelchair Feature Statistics . . . 62
4.4 ODD Checker Confusion Matrices . . . 65
4.5 Real-Time ODD Checker Performance . . . 67

5.1 Performance of VLMs . . . 82

6.1 Box-wise results on the calibration set . . . 102
6.2 Box-wise results on the test set . . . 102
6.3 Effect of α on box-wise conformalization (Calibration set) . . . 104
6.4 Effect of α on box-wise conformalization (Test set) . . . 104
6.5 Image-wise results at $\alpha = 0.1$ (Calibration set) . . . 108
6.6 Image-wise results at $\alpha = 0.1$ (Test set) . . . 108
6.7 Image-wise results at $\alpha = 0.3$ (Calibration set) . . . 110
6.8 Image-wise results at $\alpha = 0.3$ (Test set) . . . 110

7.1 Dataset Information. . . . 124
7.2 HSP dataset results . . . 131
7.3 HSP dataset runtime . . . 132
7.4 VP dataset results . . . 133
7.5 VP dataset runtime . . . 134
7.6 MNIST dataset results . . . 134
7.7 MNIST dataset runtime . . . 135
7.8 Adult dataset results . . . 136
7.9 Adult dataset runtime . . . 137
7.10 Illustrative rules (HSP dataset) . . . 138
7.11 Computational overhead (VP dataset) . . . 139
7.12 Effects of DP (VP dataset) . . . 139
7.13 DNN size metrics . . . 141
7.14 G-Explainer model size . . . 142
7.15 S-TREPAN size metrics . . . 142
7.16 Tree and model size metrics . . . 143

Notations

$\mathbb{R}$	set of real numbers
$\emptyset$	empty set
(X_i, Y_i)	input–output pair in dataset
P_{XY}	joint distribution of inputs and outputs
$\hat{f}$	trained predictor (model)
$s(X, Y)$	nonconformity score for candidate pair (X, Y)
$D_{\text{train}}, D_{\text{cal}}, D_{\text{test}}$	training, calibration, and test sets
R_i	nonconformity score of calibration sample i
$q_{1-\alpha}$	conformal quantile at level $1 - \alpha$
$C_\alpha(X)$	conformal prediction set for input X
α	miscoverage rate (user-specified error tolerance)
$1 - \alpha$	nominal coverage level
Y^k	ground-truth bounding box of object k
$\hat{Y}^k$	predicted bounding box of object k
B^k	set of pixels inside bounding box Y^k
b_w, b_h	predicted width and height of a bounding box
$R^k, R^k_{\max}$	residuals and max-residual score for box k
Δ_x, Δ_y	conformal margins in x and y directions
τ	image-level conformal margin parameter
$\widehat{C}_\tau(X)$	union of conformalized predictions for image X
$L_i(\tau)$	loss for image i under margin τ
$L_i^{\text{box}}(\tau)$	box recall loss
$L_i^{\text{pixel}}(\tau)$	pixel recall loss
n_i	number of ground-truth objects in image X_i
$A(\cdot)$	pixel area function
Q	total number of clients in federated learning
M	number of selected clients per round
N	number of communication rounds
B	number of samples per client
d	number of features (input dimension)
q	number of categories (labels)
N_{SR}	number of synthetic samples per round
$x_i \in X$	feature vector of sample i
$y_i \in Y$	ground-truth label of sample i

$\hat{y}_i$	predicted label for sample i
r_k	k-th decision rule extracted from a decision tree
$R(r_k)$	rule relevance of rule r_k
$C(r_k)$	covering (true positive rate of rule r_k)
$E(r_k)$	error (false positive rate of rule r_k)
η	rule relevance threshold
ϵ	differential privacy parameter
var_{dp}	variance perturbed with Laplace noise for DP
$GbMin, GbMax$	global minimum and maximum feature values
$GbRuleSet$	aggregated global rule set
$GbSamples$	global synthetic dataset generated at server

Acronyms

AAS	Assistive Autonomous Systems
AD	Autonomous Driving
ADAS	Advanced Driver-Assistance Systems
ADS	Autonomous Driving Systems
ADL	Adult dataset
AI	Artificial Intelligence
AV	Autonomous Vehicles
CART	Classification And Regression Trees
CFR	Cumulative Feature Ranking
CLIP	Contrastive Language-Image Pre-Training
CP	Conformal Prediction
CPS	Cyber-Physical System
DNN	Deep Neural Network
DP	Differential Privacy
DT	Decision Tree
ERL	Evolutionary Rule Learning
FedAvg	Federated Averaging
FedProx	Federated Proximal Optimization
FL	Federated Learning
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FFLA	Federated Fuzzy Learning Algorithm
GBDT	Gradient Boosted Decision Trees
GDPR	General Data Protection Regulation
GLOR-FLEX	Global-Local Rule-Based eXplanations for Federated Learning
GPT-4	Generative Pre-trained Transformer 4
GPT-4V	GPT-4 with Vision
G-Explainer	Global-Explainer
HSP	Hospital dataset
HFL	Horizontal Federated Learning
HOLDA	Hierarchical crOss-siLo feDerated Averaging
IoT	Internet-of-Things
IoU	Intersection over Union
MCC	Matthews Correlation Coefficient
ML	Machine Learning

MNIST	Modified National Institute of Standards and Technology dataset
NLP	Natural Language Processing
NN	Neural Network
non-IID	non-Independent and Identically Distributed
IID	Identically Distributed
ODD	Operational Design Domain
PFL	Personalized Federated Learning
PS	Parameter Server
REXASI-PRO	REliable and eXplainable Swarm Intelligence for People with Reduced mObility
SGD	Stochastic Gradient Descent
SOTIF	Safety Of The Intended Functionality
S-TREPAN	Server-TREPAN
STC	Sparse Ternary Compression
TLS	Transport Layer Security
TN	True Negative
TP	True Positive
UQ	Uncertainty Quantification
VFL	Vertical Federated Learning
VLM	Vision-Language Model
VP	Vehicle Platooning dataset
XAI	eXplainable Artificial Intelligence

CHAPTER 1

Introduction

1.1 eXplainable Artificial Intelligence (XAI)

The rapid integration of Artificial Intelligence (AI) technologies into everyday life has transformed how decisions are made in domains as diverse as medicine, finance, transportation, and public services. Systems driven by Machine Learning (ML) now influence credit approvals, medical diagnoses, and the behavior of Autonomous Vehicles (AV), shaping outcomes with direct societal impact. Yet, the growing power of these models comes with an equally pressing challenge: they often operate as opaque *black boxes*, producing outputs without revealing the reasoning behind them. This opacity poses fundamental risks, as decisions made by complex architectures such as Deep Neural Network (DNN)s may be accurate on average but remain unaccountable, potentially biased, and difficult to trust in critical situations. In safety-sensitive settings, particularly in Autonomous Driving Systems (ADS), such lack of transparency can undermine reliability, public acceptance, and ultimately safety itself [4]. Addressing this gap has given rise to the field of eXplainable Artificial Intelligence (XAI), which develops methods to render AI systems more interpretable, transparent, and trustworthy. The objectives of XAI extend beyond technical insight: they encompass enabling humans to understand model reasoning, providing justifications for outcomes, detecting and mitigating bias, and ensuring compliance with ethical and legal frameworks. In this context, the term trustworthy-by-design refers to AI systems in which properties such as transparency, accountability, robustness, and privacy are embedded directly into the system architecture and learning process, rather than being imposed retrospectively through post-hoc analysis or external safeguards. In essence, XAI is central to bridging the divide between increasingly complex computational models and the human need for clarity, accountability, and trust in automated decision-making [4, 5].

Different taxonomies have been proposed to organize XAI approaches. A central distinction lies between intrinsic (white-box) and post-hoc (black-box) explainability. White-box models are inherently interpretable due to their simple structure. Classical

models such as linear regression, Decision Tree (DT), and rule-based systems fall into this category. For instance, a DT expresses knowledge through a hierarchical structure of conditions, making its reasoning process transparent and easy to follow. Similarly, rule-based models describe predictions using explicit if–then rules, which can be directly interpreted by experts and non-experts alike. Black-box models, in contrast, include Neural Network (NN), ensembles such as Gradient Boosted Decision Trees (GBDT), and other highly complex algorithms whose internal mechanisms are opaque. These require post-hoc techniques to approximate or visualize their decision-making behavior.

Another important distinction is between model-specific and model-agnostic approaches. Model-specific methods exploit the internal structure of a given model class, such as analyzing feature weights in linear models or splits in DTs. Model-agnostic methods treat the model as a black box and instead rely on input–output behavior to generate explanations. Notable examples include Local Interpretable Model-Agnostic Explanations (LIME), which learns a simple surrogate model around a single prediction to highlight the features driving that decision [6], and SHapley Additive exPlanations (SHAP), which applies cooperative game theory to assign contributions of each feature toward the output [7]. Explanations may also differ in scope: global interpretability seeks to explain the overall logic of the model, such as which features are most influential across the dataset, while local interpretability focuses on explaining individual predictions, answering the question of why the model produced a specific outcome. Techniques like LIME, SHAP, and counterfactual explanations [8] are representative examples of local interpretability.

Explainability must also be understood as a lifecycle property that spans both development and deployment. Dwivedi et al. [4] emphasize a distinction between the understanding phase and the explaining phase. During the understanding phase, developers and data scientists seek interpretability to debug errors, ensure robustness, and detect potential data biases. For example, DTs or feature-importance analysis can reveal whether a model relies on spurious correlations. During the explaining phase, which occurs at deployment, end-users, businesses, and regulators require explanations to make informed decisions, justify outcomes, and ensure compliance. In domains such as insurance or healthcare, predictions without justifications are often unacceptable from ethical and legal perspectives [9].

The landscape of XAI methods is broad. Feature-based techniques identify and quantify the influence of individual input variables, for example through feature importance scores [10, 11] or partial dependence plots [12, 13], which visualize marginal effects of features on predictions. Example-based techniques instead focus on instances: counterfactual

explanations suggest minimal changes in input features that would alter a prediction [8, 14], while Anchors provide high-precision if–then rules that capture local decision boundaries of a model [15]. Surrogate models approximate black-box predictors with interpretable ones, such as DTs, to provide global-level transparency [16]. In addition, neuro-symbolic approaches [17, 18] aim to combine the statistical power of NNs with the structured reasoning capabilities of symbolic systems. Among these approaches, DTs and rule-based learners remain particularly attractive in contexts where transparency is non-negotiable. They offer concise and human-readable explanations that align well with the requirements of safety-critical systems, while also serving as effective surrogates for complex models.

Despite the progress of recent years, several challenges remain open in XAI. The most prominent is the trade-off between accuracy and interpretability: while simpler models are easier to understand, they often do not reach the accuracy of complex black-box models. Furthermore, explanations can expose underlying dataset or model biases, but ensuring fair and unbiased predictions requires additional interventions. Many XAI methods, such as SHAP, are also computationally expensive, limiting their applicability to large-scale or real-time systems. Finally, the evaluation of explanations remains a persistent issue, as human-centered metrics such as trust and usability do not always align with algorithmic criteria [9].

1.2 Video Analytics in Automated Driving Systems

The development of Advanced Driver-Assistance Systems (ADAS) and fully Autonomous Driving Systems (ADS) has become one of the most transformative trends in modern mobility. These systems aim to reduce human error, enhance safety, and improve transportation efficiency by relying on AI to perceive the environment, interpret complex situations, and make driving decisions. According to the SAE taxonomy, driving automation spans from level 0 (no automation) to level 5 (full automation), where perception forms the foundational capability enabling higher autonomy levels [19]. Without reliable perception, subsequent tasks such as planning, decision-making, and control cannot operate safely or effectively, underscoring perception as the “eyes and ears” of Autonomous Driving Systems (ADS).

Perception in ADS relies on a variety of sensor modalities, including cameras, LiDAR, radar, ultrasonic sensors, and GPS/IMU for localization [20]. Each modality has complementary strengths: cameras provide rich semantic and texture information, LiDAR

offers accurate 3D geometry, radar is robust to adverse weather, and ultrasonic sensors assist with near-field detection. Sensor fusion has therefore become a standard practice to combine these modalities and improve robustness [21]. Nevertheless, among all modalities, cameras remain the most widely adopted due to their low cost, high resolution, and compatibility with computer-vision and deep-learning pipelines [22, 23]. Consequently, video analytics has emerged as the cornerstone of modern perception systems, powering applications from lane detection and traffic-sign recognition to pedestrian and wheelchair detection.

The camera-centric perception paradigm has been accelerated by breakthroughs in DNN-based computer vision. Convolutional Neural Networks (CNNs) and transformer-based architectures have substantially advanced the accuracy of object detection, semantic segmentation, and scene understanding [24]. Object-detection networks such as Faster R-CNN, YOLO, and SSD recognize pedestrians, cyclists, and vehicles in real time [25, 26]. Semantic-segmentation models such as DeepLab [27] and ENet [28] provide pixel-level understanding for lane and drivable-area detection, while multi-object tracking algorithms [29] support dynamic awareness of surrounding actors. These advances position video analytics as a critical enabler of perception in both ADAS and higher-level ADS.

While the automotive domain remains the dominant application of ADS technologies, similar challenges arise in assistive autonomous mobility. Autonomous wheelchairs and service robots operating in structured indoor environments such as hospitals, airports, and museums perform the dynamic driving task (DDT) within a constrained Operational design domain (ODD). These devices constitute a specialized class of ADS: they operate at lower speeds and over shorter ranges but must handle dense, human-centric layouts and frequent close-range interactions. The same video-analytics foundations therefore apply, with particular emphasis on safety, comfort, and social compliance in confined environments [19, 23, 22].

In these contexts, perception failures directly affect vulnerable users, where risks include collisions in cluttered corridors, unsafe navigation through dynamic crowds, or failure to detect hazards in constrained layouts. Reliable video-based analytics are therefore essential not only for autonomy but also for user safety, comfort, and trust. Video analytics thus serves as a unifying backbone across both road-based autonomy and assistive mobility, providing the foundation for trustworthy perception in diverse application domains.

Despite considerable progress, camera-based perception faces persistent challenges. Environmental variability rain, fog, snow, and nighttime conditions can severely degrade image quality and downstream predictions; occlusion and crowding in indoor or urban

environments complicate detection and tracking; and cameras are inherently sensitive to illumination changes, with performance deteriorating under glare or poor lighting, motivating complementary sensing (e.g., radar or LiDAR) [22, 23]. Beyond robustness, real-time efficiency is vital: perception pipelines must operate under tight latency constraints to guarantee safety. Furthermore, trustworthiness remains an open issue, as DNN-based video models are often opaque, miscalibrated, or biased, complicating verification and raising barriers to regulatory acceptance [30]. These limitations motivate frameworks that combine video analytics with Uncertainty Quantification (UQ), runtime monitoring, and explainability, establishing the methodological background for the developments presented in this thesis.

1.3 Federated Learning (FL)

The increasing importance of data privacy and the emergence of strict regulations such as the General Data Protection Regulation (GDPR) have exposed limitations of conventional centralized ML. In traditional pipelines, data from multiple organizations or devices is aggregated in a central repository, where it is processed and used to train models. While effective for producing high-performance models, this approach creates significant risks: sensitive information may be leaked during transfer or storage, organizations may be unwilling or legally unable to share their data, and centralizing massive datasets introduces security vulnerabilities [31]. These concerns are particularly pressing in domains such as healthcare, finance, and ADS, where both the scale and sensitivity of data are high.

Federated Learning (FL) has emerged as a paradigm to address these challenges. First introduced by Google in 2016 in the context of mobile keyboard prediction, FL enables multiple clients such as mobile devices, vehicles, or institutions to collaboratively train a shared global model without exchanging their raw data [32]. Instead, each client keeps its data locally, performs training on its own device, and shares only model updates with a coordinating PS. The server aggregates these updates and broadcasts the improved global model back to participants. Through this iterative process, the model benefits from distributed data while preserving privacy [33]. The overall workflow of FL is illustrated in Figure 1.1.

Beyond privacy preservation, FL is increasingly studied within the broader context of *Trustworthy AI*. Trustworthy AI refers to the design of intelligent systems that satisfy a set of fundamental requirements. These commonly include privacy and data protection,

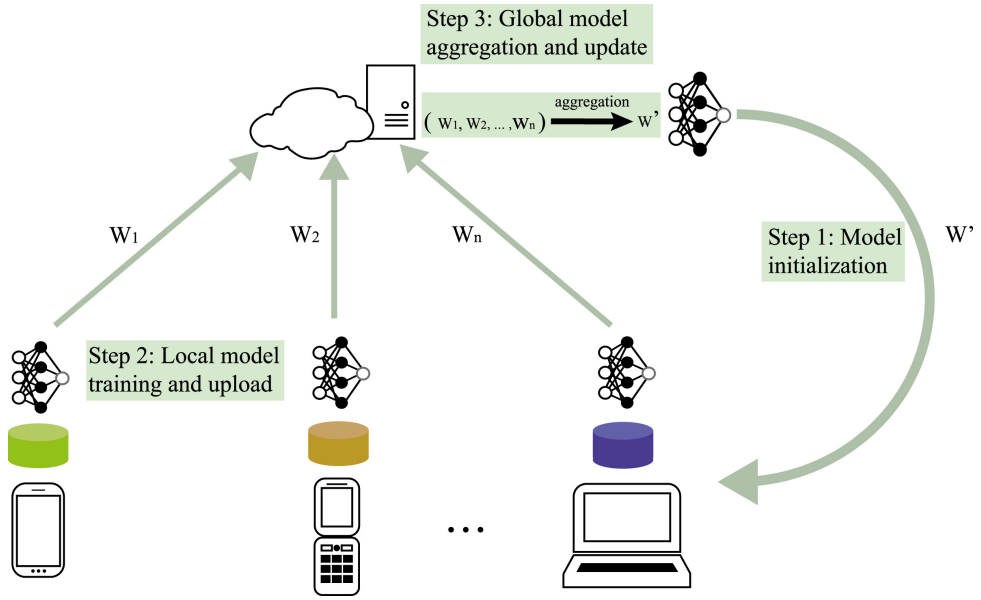


Figure 1.1: Typical FL workflow. Clients train local models on their private data and send only parameter updates ($W_1, W_2, \ldots, W_n$) to a central server, which aggregates them (e.g., using FedAvg) into a global model W'. The updated global model is then redistributed back to clients.

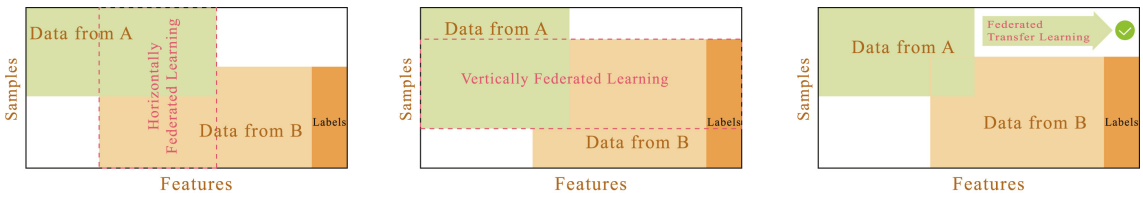
robustness and security, transparency and explainability, fairness, and accountability. In distributed learning settings, these requirements become tightly coupled: models must be trained collaboratively without exposing sensitive data, remain robust to unreliable or malicious participants, and support mechanisms for auditing and verification of global behavior. FL therefore represents not only a privacy-preserving training paradigm, but also an architectural foundation upon which broader trustworthiness properties such as robustness, accountability, and explainability must be explicitly engineered.

In federated settings, trustworthiness requirements extend beyond data privacy alone. The distributed and heterogeneous nature of FL systems complicates transparency, accountability, and robustness, particularly when deployed across large-scale Internet-of-Things (IoT) and edge environments. Models trained under extreme non-IID conditions may exhibit unstable or opaque behavior that is difficult to audit or justify, especially when decisions affect safety-critical or regulated applications. In this context, explainability is not merely a diagnostic tool but a foundational requirement for enabling traceable decision-making, system verification, and accountability across decentralized participants. Consequently, there is growing interest in FL architectures that embed interpretability and

robustness directly into the learning process, rather than relying solely on post-hoc explanations applied to black-box models.

The distinction between centralized and FL is therefore fundamental. In centralized systems, raw data flows toward the server, raising privacy and compliance concerns. In FL, only abstracted knowledge (parameters or gradients) is exchanged, while the sensitive data itself remains on-device [31]. This shift not only reduces legal and ethical risks but also leverages the growing computational capacity of edge devices within the IoT, enabling large-scale collaborative intelligence across distributed infrastructures.

Different categories of FL have been proposed depending on how data is distributed. Horizontal Federated Learning (HFL) applies when different clients share the same feature space but have distinct user samples; the Google keyboard is a canonical example.Vertical Federated Learning (VFL) is used when organizations possess different features about the same set of users, such as banks holding financial attributes and e-commerce platforms recording purchase histories. Federated Transfer Learning becomes relevant when neither user samples nor features overlap significantly, allowing knowledge from auxiliary tasks to improve learning in data-sparse domains [31, 33]. These main categories of FL are depicted in Figure 1.2. Cross-device and cross-silo deployments represent two practical settings: in the former, millions of personal devices participate intermittently, while in the latter, a smaller number of stable organizations (e.g., hospitals or banks) contribute throughout training.



(a)Horizontal Federated Learning (b)Vertical Federated Learning (c)Federated Transfer Learning

Figure 1.2: Main categories of FL according to data distribution. (a) Horizontal FL: clients share the same feature space but different samples. (b) Vertical FL: clients share the same samples but different feature sets. (c) Federated Transfer Learning: little overlap in both features and samples, requiring transfer learning techniques.

Despite its promise, FL introduces its own set of challenges. Communication overhead is a major bottleneck: the frequent exchange of large model updates between clients and servers can dominate training time and energy consumption. Compression methods such as Sparse Ternary Compression (STC) and asynchronous update protocols have been proposed to mitigate this issue [31]. System and data heterogeneity further complicate

matters: edge devices vary widely in computational resources, connectivity, and reliability, while the local data distributions are often non-IID, making model convergence more difficult compared to centralized training [33]. Optimizers such as Federated Proximal optimization (FedProx) [34] attempt to address some of these issues by adapting aggregation to heterogeneous environments. These challenges directly impact the trustworthiness of federated systems, as failures in robustness, security, or accountability can undermine confidence in the learned global models, particularly in safety-critical and regulated domains.

Beyond efficiency and convergence, privacy and security constitute additional and tightly coupled trustworthiness challenges. Although raw data never leaves the client, adversaries may still infer sensitive information from shared model updates through membership inference attacks, poison the training process by manipulating local data or parameters, or introduce backdoors into the global model [35]. Common defenses include secure aggregation, homomorphic encryption, and the incorporation of noise via Differential Privacy (DP), though these safeguards often introduce trade-offs in terms of computational overhead or model accuracy [31]. Furthermore, the distributed nature of FL complicates accountability, as identifying malicious clients or enforcing fairness across participants remains non-trivial. FL represents a paradigm shift in the design of privacy-preserving, distributed ML systems. By enabling collaborative model training without centralized data collection, it directly addresses the tension between data utility and data protection. At the same time, it raises new technical challenges in optimization, communication, security, and interpretability that continue to drive an active research agenda. Addressing these challenges is essential for positioning FL as a practical enabler of trustworthy AI, particularly in safety-critical and privacy-sensitive domains. This thesis builds on these challenges by proposing an interpretable FL framework that directly addresses trustworthiness requirements beyond privacy alone.

1.4 Contributions of this Thesis

The central theme of this thesis is the integration of XAI principles into AI systems across two domains: perception through video analytics in autonomous and assistive mobility, and distributed training in FL with a particular focus on interpretability, accountability, and robustness under data heterogeneity. While advances in DNN-based perception and federated optimization have yielded remarkable performance, their opacity poses substan-

tial barriers to trust, accountability, and regulatory acceptance in safety-critical settings. The contributions presented in this thesis address this challenge by embedding explainability as a design principle rather than a post-hoc addition, ensuring that transparency and reliability are intrinsic to the system.

The first line of contributions is situated within the context of the REXASI-PRO project*, a European initiative aimed at developing greener and trustworthy-by-design AI solutions for assistive platforms such as autonomous wheelchairs and drones. The project's objective is to enable safe and efficient navigation in structured indoor environments such as hospitals, airports, and museums. The envisioned system integrates an autonomous wheelchair with onboard cameras and a supporting drone (Figure 1.3), while fixed cameras installed in the environment provide complementary perspectives (Figure 1.4). Together, these enable multi-view perception for risk-aware navigation.



Figure 1.3: Deployment scenario in the REXASI-PRO framework: collaboration between an autonomous wheelchair and a drone for safe indoor navigation.

Within this framework, the thesis advances a dual-stage approach to video analytics. The first stage focuses on establishing a perception backbone by fine-tuning state-of-the-art object detectors with domain-specific categories such as the wheelchair, thereby extending conventional detection capabilities to assistive mobility contexts. The second stage integrates explainability and UQ into perception pipelines. To this end, an ODD Checker was designed to provide interpretable monitoring of prediction certainty. By

*Demonstration video: `https://www.youtube.com/watch?v=u3od-cBPD0I`

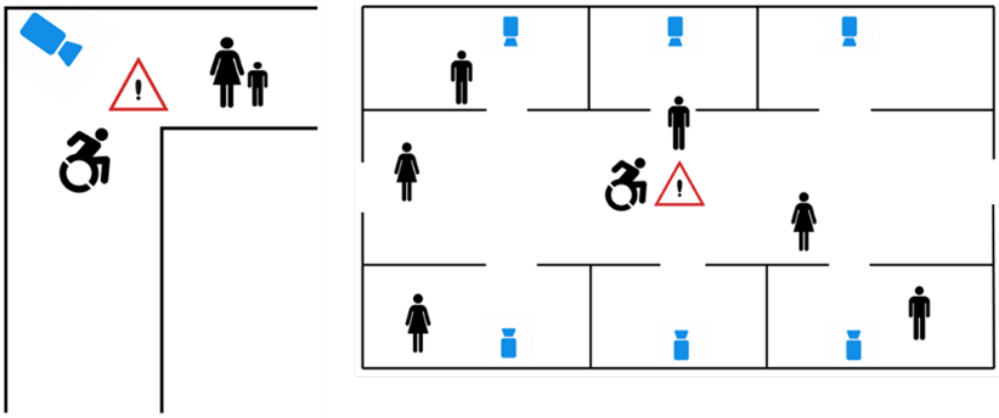


Figure 1.4: Risk-aware perception in assistive mobility: vision-based analytics detect safety-critical situations (e.g., occluded hazards, crowding) and support runtime monitoring for trustworthy decision making within REXASI-PRO.

combining visual feature analysis with DT-based rules, the Checker establishes explicit and auditable decision boundaries, enabling operational safety monitoring and human-understandable justifications.

Explainability is then extended beyond individual object predictions to holistic scene-level reasoning. A benchmarking study of Vision-Language Models (VLMs), including CLIP, MiniGPT-4, and GPT-4V, was conducted to assess their ability to classify indoor navigation scenes as either "Safe to Proceed" or "Risky to Proceed." Dimensions included obstacle proximity, crowd density, visibility degradation, and stair-related hazards. This contribution demonstrates how multimodal architectures can provide semantically rich, interpretable reasoning in human-centric environments and introduces conceptual tools, such as the swan taxonomy, to characterize rare and ambiguous risk conditions that challenge traditional perception systems.

In addition, the thesis contributes to principled UQ by applying CP to object detection pipelines. By comparing box-wise and image-wise conformalization strategies, the work establishes methodological insights into how distribution-free statistical guarantees can be incorporated into safety-critical perception.

While the first set of contributions focuses on perception, explainability, and uncertainty in centralized and embedded vision systems, the second line of contributions shifts attention to distributed intelligence and collaborative learning under privacy and data heterogeneity constraints. Specifically, this thesis addresses trustworthy distributed intelligence through FL. While existing federated optimization frameworks such as FedAvg and FedProx primarily focus on privacy preservation through data locality, they rely on black-

box neural models that offer limited transparency, accountability, and robustness under heterogeneous data distributions. This thesis addresses these limitations by introducing FL-IRT (Federated Learning for Interpretable Rule Trees), a FL framework that embeds trustworthiness principles directly into the learning process. More broadly, this work contributes a FL framework that operationalizes multiple pillars of Trustworthy AI within a single, coherent system. FL-IRT integrates privacy preservation, intrinsic explainability, robustness to extreme non-IID data distributions, and computational efficiency by design. By adopting lightweight, rule-based models and intrinsic (ante-hoc) interpretability, the framework avoids the overhead and opacity of black-box neural approaches while enabling fast training, low communication cost, and auditable global behavior. This system-level perspective positions FL-IRT as a practical and trustworthy FL solution for resource-constrained, distributed, and edge-based environments.

From an interpretability and accountability perspective, FL-IRT adopts DTs as its core learning paradigm, enabling native transparency without reliance on post-hoc explanation techniques. Human-readable decision rules are learned locally at client devices and subsequently aggregated at the server into a globally consistent and auditable rule set, providing traceable decision logic suitable for regulated and safety-critical domains.

In terms of privacy and security, FL-IRT adheres to the FL paradigm by ensuring that raw client data remain local. This is complemented by secure client–server communication via Transport Layer Security (TLS) and by client-side enforcement of DP through noise injection mechanisms applied to local statistics and rule relevance measures, limiting the risk of sensitive information leakage from shared updates.

Robustness and fairness are addressed through mechanisms specifically designed for statistically heterogeneous (non-IID) environments. The framework incorporates confidence-aware sampling strategies and data-range conditions that enable the recovery of globally relevant features even when they are absent or weakly represented in individual client models. This design supports stable aggregation across diverse clients and mitigates the impact of distributional skew on global model behavior.

Finally, the use of lightweight, rule-based models contributes to computational efficiency and system reliability, making FL-IRT suitable for deployment in resource-constrained and edge-based settings. Together, these design choices position FL-IRT as a trustworthy-by-design FL framework, aligning privacy, robustness, transparency, and accountability within a unified distributed learning architecture.

Taken together, these contributions position the work within the broader contexts of XAI, video analytics, and FL. They establish a coherent research agenda in which

explainability serves as the common thread across distinct AI paradigms. From object detection and scene-level reasoning in video analytics, to principled UQ in perception, and interpretable rule-based frameworks in FL, the thesis demonstrates that transparency can be embedded at the core of system design. By aligning methodological innovation with practical evaluation, the work advances the development of trustworthy-by-design AI systems capable of safe and accountable operation in safety-critical domains such as autonomous and assistive mobility.

1.5 Structure of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 surveys the state of the art, reviewing previous work on XAI, video analytics in perception systems, scene understanding, UQ, and explainability in FL. Chapter 3 introduces the methodological foundations of perception and vision in ADS, including the development and fine-tuning of a custom object detector. Chapter 4 presents the design and evaluation of the ODD Checker for interpretable monitoring of video analytics in assistive mobility scenarios. Chapter 5 extends explainability to the scene level by benchmarking VLM for safety assessment in autonomous wheelchair navigation. Chapter 6 explores CP as a framework for UQ, establishing statistical guarantees on perception outputs. Chapter 7 turns to distributed learning, presenting FL-IRT as an interpretable FL framework. Finally, Chapter 8 concludes the thesis by synthesizing the findings.

CHAPTER 2

Previous work

This chapter reviews state-of-the-art applications of eXplainable Artificial Intelligence (XAI) in two domains central to this thesis: video analytics for autonomous and assistive mobility and Federated Learning (FL) under privacy and data-heterogeneity constraints. The survey spans advances in object detection, interpretable monitoring, scene-level reasoning with Vision-Language Model (VLM), Uncertainty Quantification (UQ) through Conformal Prediction (CP), and explainable federated frameworks. These works establish the methodological background for embedding explainability as a core design principle in perception and distributed learning.

2.1 Perception and Vision in Autonomous Driving Systems (ADS)

Object detection has become one of the most extensively studied topics in computer vision, forming the basis of perception in autonomous systems, medical imaging, and surveillance. Early work relied on hand-crafted features such as Histogram of Oriented Gradients (HOG) [36] and Deformable Part Models (DPM) [37], which provided limited robustness under real-world conditions. The advent of CNNs marked a turning point. Region-based detectors, including R-CNN [38], Fast R-CNN [39], and Faster R-CNN [40], introduced a two-stage pipeline that combined region proposals with CNN-based classification and regression, achieving unprecedented accuracy [38, 40]. Despite their success, these models were computationally expensive, constraining their use in real-time scenarios.

To overcome this bottleneck, single-stage detectors were developed, predicting class labels and bounding boxes in a single pass. The YOLO family (You Only Look Once) [41] and the SSD framework (Single Shot MultiBox Detector) [42] demonstrated that real-time detection was possible without severe accuracy trade-offs. These models, particularly YOLO, became foundational for applications requiring low-latency perception. Over

successive versions, YOLO integrated improvements such as anchor-free bounding boxes, optimized feature aggregation, and advanced data augmentation, positioning it as the most widely deployed real-time detector. More recently, transformer-based architectures such as DETR (Detection Transformer) [43] have introduced end-to-end detection pipelines based on attention mechanisms, eliminating hand-crafted components. However, their computational demands and training complexity currently hinder adoption in resource-constrained or safety-critical systems.

While large-scale pretrained models (e.g., those trained on the COCO dataset [44]) serve as strong baselines, their generalization to domain-specific tasks is often limited. Many safety-critical applications involve rare categories or specialized conditions absent from standard benchmarks. Transfer learning through fine-tuning has thus become the prevailing strategy to adapt pretrained models to new domains. Pretrained weights are commonly used as initialization, with early layers frozen to preserve generic features, while higher layers and detection heads are retrained to capture task-specific semantics [45, 46].

Recent research has also highlighted limitations. Li et al. [47] identified a misalignment between standard pre-training, which emphasizes image-level classification, and downstream object detection tasks, which require instance-level semantics and precise localization. They proposed AlignDet, which decouples pre-training into image-domain and box-domain objectives, enabling detectors such as Faster R-CNN, RetinaNet, and DETR to transfer semantic knowledge more effectively. Experiments confirmed that aligning pre-training tasks with fine-tuning objectives yields consistent improvements in mean Average Precision (mAP) under domain shifts. This line of work underscores that fine-tuning is not only about adapting weights but also about designing pre-training strategies compatible with detection requirements.

The versatility of fine-tuning is particularly evident in safety-critical applications where annotated data is scarce but accuracy is essential. In medical imaging, for example, Montalbo et al. [48] fine-tuned a YOLO-based model for brain tumor detection on MRI scans, achieving competitive performance despite limited training data. Similarly, in surveillance and mobility contexts, fine-tuned detectors have been shown to adapt to new lighting conditions, camera perspectives, and adverse weather, outperforming generic pretrained baselines.

Despite this progress, most prior work has focused on conventional benchmarks, with limited attention to rare or domain-specific categories that are critical in real-world ADS. In particular, objects such as wheelchairs essential for assistive mobility scenarios are poorly represented in standard datasets. This thesis addresses this gap by fine-tuning

YOLOv8 for custom safety-critical categories, providing a perception backbone that supports downstream explainability and monitoring mechanisms.

2.2 Extending and Monitoring the Operational Design Domain in Autonomous Driving Systems

The concept of ODD for ADS has been extensively studied, particularly in the context of object detection. Several works [49, 50, 51] have centered on improving the robustness of models under adversarial settings, such as data poisoning and adversarial attacks. For example, Sun, Chen, et al. [52] recently proposed a robust learning methodology to extend the ODD of perception modules in ADS, addressing performance degradation under challenging real-world conditions such as variations in weather and lighting. Additionally, advancements in domain adaptation have been instrumental in improving visual perception performance. Adversarial feature alignment [53, 54], image-level generative mapping [55, 56], and pseudo-label self-training [57, 58] have been used to address the challenges posed by domain shifts. While these approaches have made significant strides in extending the ODD and improving robustness, they often focus on improving model performance through training strategies, rather than providing real-time, interpretable assessments of operational domain compliance. This gap highlights the importance of interpretable monitoring, which aims to ensure transparency and trust in the system's predictions during deployment.

Research on interpretable monitoring in ADS spans multiple domains, including perception and planning/prediction, each addressing distinct facets of safety and decision-making. In the case of perception, the goal of interpretable monitoring is to verify the decision-making processes of object detection systems through interpretable models and visual explanations. Kronenberger et al. [59] introduced interpretable DNNs for traffic sign recognition, integrating visual concepts such as colors, shapes, and symbols to verify network decisions. Similarly, Hacker et al. [60] proposed a monitor using saliency maps computed through occlusion sensitivity [61], enabling real-time comparisons with pre-computed category maps. Keser et al. [62] developed a model-agnostic monitor leveraging concept bottleneck models to conduct plausibility checks.

Interpretable monitoring for planning and prediction in ADS guarantees safety at the critical phase of decision making. Symbolic representations, such as DTs, are often employed to provide interpretable safety assessments [63, 64, 65]. These models generally try

to predict binary or scalar safety values for high-level actions, such as collision avoidance and lane changes, based on state descriptions but without visual cues. To address this limitation, other works integrate perception data to enhance decision-making processes. For instance, Schmidt et al. [66] introduced a pipeline leveraging DTs extracted from reinforcement learning agents, ensuring both interpretability and formal safety guarantees through logical rule sets. Bao et al.'s DRIVE model [67] integrates attention-guided reinforcement learning for accident anticipation, linking predictions to visually explainable regions in dashcam videos. Gilpin et al. [68] designed an anomaly detection framework employing a hierarchy of explanations that synthesizes outputs from multiple subsystems, reconciling inconsistencies through logical rules and prioritizing safety. Reducing misclassification error, either by lowering false positive rates or false negative rates, has also been recognized as an important step toward improving reliability, making interpretable monitoring more trustworthy in operational settings [69, 70].

Although existing methods significantly advance interpretability and safety assurance, they predominantly focus on high-level tasks such as accident or collision prediction. In contrast, this thesis approaches the problem from a complementary perspective, focusing on low-level monitoring of object detection outputs to assess prediction certainty. By formalizing the ODD as a set of operational conditions derived from visual feature analysis, the proposed approach enables real-time identification of prediction uncertainty without relying on internal model features. This integration of visual analysis and interpretable rule-based monitoring offers a lightweight and transparent framework that complements existing efforts by supporting safety assurance directly at the perception layer.

2.3 Scene Comprehension and Safety Monitoring with Vision-Language Model (VLM)

Recent advances in VLMs have introduced powerful multimodal reasoning capabilities that extend beyond conventional perception systems. Models such as CLIP [71], ALIGN [72], and BLIP-2 [73] demonstrated that contrastive and generative pretraining on large-scale image–text pairs enables robust zero-shot recognition and natural-language explanation without task-specific supervision. More recent architectures, including MiniGPT-4 [74] and GPT-4V [75], further integrate vision and language through large-scale generative modeling, supporting open-ended dialogue and naturalistic reasoning about complex scenes. Although originally trained for general-purpose vision–language tasks, their abil-

ity to transfer semantic knowledge across domains has stimulated research in safety-critical applications.

Within autonomous driving and robotics, VLMs have been explored as a means of enhancing scene understanding, anomaly detection, and instruction following. PaLM-E [76] and RT-2 [77] demonstrated that web-scale knowledge from VLMs can be adapted to embodied decision-making, while DriveLM [78] and Talk2BEV [79] incorporated language-driven reasoning into traffic scene analysis. These studies highlight the semantic flexibility of VLMs, but also reveal persistent challenges such as inference latency, hallucinations, and limited robustness under real-world conditions [80]. Surveys further confirm that while VLMs promise greater interpretability, their practical deployment in safety-critical systems remains difficult [81, 82, 83].

At the same time, classical scene-understanding pipelines based on semantic segmentation, affordance reasoning, and 3D reconstruction [84, 85] have shown fragility in cluttered or dynamic environments. Indoor assistive mobility platforms such as autonomous wheelchairs or service robots are particularly sensitive to these limitations: navigation in human-centric layouts requires reasoning about ambiguous conditions such as partial occlusions, high crowd density, or degraded visibility. In such contexts, VLMs offer an opportunity to complement object-level detectors with semantically rich, language-grounded reasoning that can elevate perception from isolated detections to holistic scene-level safety assessments.

Explainability further motivates the integration of VLMs into safety-critical perception. Recent approaches combine prediction with natural-language rationales [86], providing human-auditable justifications that go beyond low-level feature attributions. Risk-assessment frameworks complement this direction by recognizing that unsafe conditions cannot be exhaustively specified in advance. The swan metaphor [87] distinguishes “grey swans” (rare but partly understood hazards) from “black swans” (fundamentally unforeseeable risks), emphasizing the need for flexible, semantic monitoring tools. Within this view, VLMs are well positioned to act as safety filters, flagging borderline conditions such as ambiguous crowding or degraded visibility that might be missed by conventional detectors.

Despite these advances, most prior work has focused on outdoor driving or general robotic navigation, with little attention to assistive indoor mobility. In particular, there has been no systematic benchmarking of VLMs for explainable scene-level safety filtering in environments such as hospitals, airports, or train stations. Addressing this gap, this thesis contributes by evaluating CLIP, MiniGPT-4, and GPT-4V on curated indoor navigation

scenes, explicitly analyzing their ability to provide interpretable safety judgments and employing the swan taxonomy to characterize rare and ambiguous risk conditions.

2.4 Uncertainty Quantification through Conformal Prediction

UQ has become an essential component of reliable perception in ADS and assistive indoor mobility platforms, where safety requirements demand not only accurate predictions but also calibrated confidence estimates. Traditional approaches for UQ in object detection have largely relied on probabilistic and Bayesian formulations. Bayesian NN explicitly model parameter uncertainty [88, 89], while approximate Bayesian methods such as Monte Carlo dropout [90, 89, 91] and deep ensembles [92, 93, 94] remain popular baselines. Although effective in improving calibration, these methods typically require multiple forward passes or model replications, imposing significant computational overhead that is impractical for real-time deployment in mobility systems.

Calibration-oriented techniques have also been proposed to improve the reliability of detectors. Temperature scaling and post-hoc calibration [95, 96, 97] provide empirical improvements in classification, while extensions tailored to object detection [98] adapt these ideas to bounding-box regression. However, such approaches lack finite-sample guarantees and often degrade under distribution shifts, a common occurrence in dynamic indoor and driving environments.

To address these limitations, recent works have investigated Conformal Prediction (CP) as a distribution-free, model-agnostic framework for bounding-box calibration. Early explorations include de Grancey et al. [99], who introduced a conformal approach for object detection, demonstrating how additive and multiplicative residuals from calibration data can be used to construct bounding boxes with formal coverage guarantees. Andéol et al. [3] extended this line of work to railway signal detection, evaluating Bonferroni-corrected and max-residual strategies, and highlighting the trade-off between conservativeness and efficiency. Building on these foundations, Zhou et al. [2] formalized CP for multi-class detection tasks, making it applicable in cluttered real-world environments such as autonomous driving and assistive mobility.

A related but distinct thread of research leverages PAC-based calibration. Li et al. [100] introduced a PAC framework for multi-object detection and tracking, generalizing the guarantees of Park et al. [101] to multi-stage detection pipelines. While their approach

provides theoretical rigor, it integrates tightly into specific architectures such as Fast R-CNN, limiting direct applicability to modern one-stage detectors (e.g., YOLO or DETR). In contrast, CP operates post-hoc, requires only a calibration set, and is computationally attractive for embedded and real-time perception systems.

Surveys on uncertainty in object detection and autonomous driving [102] confirm that while Bayesian and ensemble-based methods remain dominant, CP is increasingly recognized as a promising and computationally efficient alternative with rigorous statistical guarantees. Nevertheless, existing CP methods often assume single-class or sparse-object scenes [99, 3], and image-wise extensions such as conformal risk control (Conformal Risk Control (CRC)) [103] can be overly conservative in practice. This thesis addresses this gap by applying CP to object detection with custom safety-critical categories, providing statistically valid uncertainty bounds that enhance the reliability of perception in ADS and assistive mobility scenarios.

2.5 Explainability in Federated Learning Systems

Federated Learning (FL) has emerged as a prominent paradigm for distributed ML, enabling collaborative model training while preserving data privacy and ownership. Despite these advantages, most widely adopted FL frameworks rely on DNNs trained using aggregation strategies such as FedAvg [32] and FedProx [34]. While these approaches achieve strong predictive performance, they operate as black boxes, offering limited transparency into either the training process or the resulting global model. This lack of explainability poses a significant obstacle to the adoption of FL in safety-critical and regulated domains, where transparency, accountability, and auditability are fundamental requirements.

A widely adopted conceptual distinction in the literature separates explainability in FL into *ante-hoc* and *post-hoc* approaches [104]. Ante-hoc explainability refers to transparency that is embedded directly into the system design, enabling human understanding of decisions as they are made. Post-hoc explainability, by contrast, aims to explain the behavior of trained models through external analysis methods applied after training. This distinction is particularly relevant in FL settings, where both the distributed training process and the learned global model must be understandable to foster trust among participants and stakeholders.

Ante-hoc explainability encompasses transparency at two complementary levels: the FL process itself and the intrinsic interpretability of the adopted models. At the process

level, explainability concerns whether key steps such as data preparation, local training, client selection, and aggregation are transparent, justifiable, and free from systematic bias. For example, in cross-institutional FL scenarios such as healthcare, ante-hoc explainability includes whether data preprocessing pipelines are consistent across clients and whether participation mechanisms treat institutions fairly. These properties are essential for accountable and auditable decision-making.

Research explicitly targeting explainable FL processes remains relatively limited. Existing studies primarily address explainability indirectly through transparent mechanisms for feature selection, sample selection, or client selection. Filtering-based federated feature selection methods identify informative features using model-agnostic criteria, offering clearer interpretability than wrapper or embedded approaches [105, 106, 107]. Similarly, explainable federated sample selection strategies rely on intuitive metrics such as loss values, prediction confidence, or gradient norms to identify influential samples [108, 109, 110, 111]. Explainable client selection further enhances transparency by explicitly evaluating client utility based on statistical or system-level criteria, enabling auditable participation decisions [112, 113, 114, 115, 116, 117]. While these approaches improve transparency at specific stages of FL, they do not address the interpretability of the global model itself.

Beyond process-level considerations, ante-hoc explainability is strongly influenced by the intrinsic interpretability of the learning models. Conventional models such as linear models, rule-based systems, naïve Bayes classifiers, and Decision Trees (DTs) naturally support transparent reasoning, in contrast to black-box DNNs. Several works have explored the integration of such models into FL frameworks, although adaptations are often required to accommodate privacy constraints and heterogeneous data distributions.

Rule-based models, typically expressed as IF–ELSE rules, offer particularly high interpretability. Fuzzy rule-based approaches, including Takagi–Sugeno–Kang systems, have been adapted to FL through multi-stage collaborative procedures that preserve data locality while learning interpretable decision logic [118, 119, 120, 121]. These methods demonstrate the feasibility of interpretable federated models, though achieving global coherence across heterogeneous clients remains challenging.

Tree-based models constitute another important class of intrinsically interpretable approaches. Federated DT construction methods collaboratively compute split criteria across clients while preserving privacy [122], and extensions have been proposed for vertical FL settings [123, 124]. Ensemble-based methods, such as federated random forests and boosting algorithms, further improve performance [125, 126], but often at the

expense of global interpretability. Even when composed of interpretable components, the resulting ensemble models can be difficult to audit. In contrast, commonly used baselines such as FedAvg [32] and FedProx [34] aggregate neural parameters and offer little transparency beyond post-hoc explanations. Hybrid methods such as Global-Local Rule-Based eXplanations for Federated Learning (GLOR-FLEX) [127] partially address this limitation by extracting local rules and aggregating them at the server, yet they typically prioritize fidelity to local models rather than producing a unified, globally interpretable structure.

Post-hoc explainability provides an alternative means of understanding FL systems by analyzing model behavior after training. These approaches are particularly common in DNN-based FL frameworks, as they do not require modifications to the underlying architecture. In collaborative scenarios, post-hoc explainability supports alignment with domain expertise, attribution of model performance to individual clients or features, and retrospective auditing.

Post-hoc methods in FL can be broadly grouped into model evaluation, importance evaluation, and visualization techniques. Federated model evaluation assesses performance across heterogeneous client datasets without centralizing test data [128, 129]. Importance evaluation methods focus on feature attribution or client contribution, often using gradient-based techniques or Shapley-value-inspired formulations [130, 131, 132, 133, 134, 135]. Visualization techniques further enhance transparency by presenting learning dynamics, client participation, and system status in an interpretable manner, as supported by several research and industrial FL frameworks [136, 137, 138, 139].

Overall, existing explainability approaches in FL remain fragmented. DNN-based frameworks such as FedAvg and FedProx offer limited transparency, while interpretable model-based approaches often struggle with scalability, robustness under non-IID data, or the construction of coherent global models. These limitations highlight the need for FL frameworks that embed interpretability natively into the learning process, rather than relying solely on post-hoc explanations. These observations directly motivate the development of FL frameworks that combine intrinsic model interpretability with robustness to data heterogeneity while preserving privacy guarantees.

2.6 Summary and Research Gaps

This chapter reviewed prior work on explainable AI across two domains addressed in this thesis: video analytics for autonomous and assistive mobility, and FL. While substantial progress has been made in model accuracy, robustness, and scalability, several important research gaps remain. These gaps directly motivate the contributions presented in the following chapters.

In video analytics for indoor autonomous and assistive mobility within ADS and Assistive Autonomous Systems (AAS), current object detection pipelines are constrained by the scope of standard pretrained datasets. Widely used detectors such as YOLOv8 are initialized on generic benchmarks (e.g., COCO), which omit safety-critical categories relevant to assistive scenarios, most notably wheelchairs. This limitation reduces perception completeness and introduces systematic failure modes. Furthermore, inter-class ambiguity between visually similar objects (e.g., wheelchairs, chairs, and bicycles) leads to elevated False Positive (FP) rates, undermining reliability in cluttered indoor environments. These limitations motivate the need for domain-adapted perception backbones that explicitly support safety-critical categories and enable downstream explainability and monitoring.

Beyond object detection, a major open challenge concerns the definition and monitoring of the ODD. Existing regulatory frameworks provide high-level guidance but offer limited support for real-time verification of ODD compliance in complex and dynamic environments. Although many learning-based approaches improve robustness through training strategies or domain adaptation, they rarely provide interpretable mechanisms for assessing whether current operating conditions remain within validated boundaries. This gap highlights the need for explainable, rule-based monitoring approaches capable of providing transparent and auditable safety verification during runtime.

At the scene level, recent advances in VLMs introduce new opportunities for semantic reasoning and human-aligned explanations. However, their application to assistive indoor mobility remains largely unexplored. In particular, there is a lack of systematic evaluation of VLMs as explainable safety filters under degraded or ambiguous conditions, such as crowding, occlusions, or reduced visibility. The trade-offs between reasoning capability, robustness, and computational efficiency in such environments are not yet well understood.

UQ represents another critical gap in safety-critical perception. While Bayesian and ensemble-based approaches improve calibration, they impose significant computational overhead and often lack formal guarantees under distribution shift. CP provides a distribution-free alternative with statistical guarantees, yet existing methods are typically

evaluated under simplified assumptions and sparse object settings. The applicability and limitations of object-level and scene-level conformalization strategies for indoor assistive mobility therefore remain insufficiently explored.

Finally, in FL, existing optimization frameworks primarily focus on privacy preservation through data locality, while relying on black-box DNNs that offer limited transparency and robustness under extreme non-IID data distributions. There is a lack of lightweight, intrinsically interpretable FL approaches that can simultaneously achieve competitive performance, handle severe statistical heterogeneity, and remain suitable for resource-constrained and edge-based settings. Moreover, many existing works address isolated aspects of trustworthiness rather than providing a complete system that integrates interpretability, privacy protection (e.g., DP), secure communication (e.g., TLS), and computational efficiency by design. To address this gap, this thesis develops a lightweight, ante-hoc explainable FL framework that provides intrinsic interpretability at both client and server levels, while maintaining competitive performance and robustness under extreme non-IID data distributions.

Addressing these gaps requires approaches that embed explainability, robustness to data heterogeneity, and system-level trustworthiness directly into the learning and perception pipelines, particularly for safety-critical autonomous and assistive applications.

CHAPTER 3

Perception and Vision in Autonomous Driving Systems

3.1 Introduction

This chapter establishes the perception backbone of the thesis. Within the REXASI-PRO project, state-of-the-art object detection methods are adapted and fine-tuned—most notably YOLOv8—to extend perception capabilities toward safety-critical mobility applications. By constructing a robust detection framework with domain-specific categories such as Wheelchairs and Persons, this stage provides the essential foundation upon which subsequent explainability and monitoring mechanisms are embedded.

3.1.1 Background

Object detection is a fundamental capability in Autonomous Driving Systems (ADS) and Advanced Driver-Assistance Systems (ADAS), as it enables the localization and classification of relevant objects in the environment. Reliable detection underpins high-level tasks such as planning, navigation, and safety verification, making it indispensable for both AVs and assistive mobility platforms.

While object detection has been extensively studied in outdoor driving contexts, indoor smart mobility introduces distinct perception challenges. Environments such as hospitals, stations, and airports are characterized by high pedestrian density, frequent occlusions, and visually cluttered backgrounds. Objects of interest often include not only people but also mobility aids, furniture, and service equipment. Studies in indoor scene understanding emphasize that recognition performance deteriorates compared to outdoor environments due to the higher appearance variability and clutter typical of indoor spaces [140, 141]. Furthermore, occlusion and geometric ambiguity complicate object delineation, requiring detection models that can integrate both semantic cues and spatial context.

Recent advances in deep ML, particularly convolutional neural networks, have established single-stage detectors as the standard for real-time perception. The YOLO family exemplifies this trend by achieving end-to-end detection in a single forward pass, balancing speed and accuracy [142]. Earlier models in the YOLO series, such as YOLOv4 and YOLOv5, relied on anchor-based bounding box prediction and required careful hyperparameter tuning [143]. YOLOv8, released in 2023, departed from this paradigm by introducing anchor-free detection, an optimized CSP-based backbone, and an enhanced PANet neck for multi-scale feature fusion [144]. These innovations address limitations in generalization, training stability, and scale adaptability, thereby positioning YOLOv8 as a versatile detector for both academic research and industrial deployment.

3.1.2 Open Challenges

Despite these advances, pretrained YOLOv8 models remain constrained by the scope of the datasets on which they are initialized. Models trained on generic datasets such as COCO [44] do not cover all object categories relevant to indoor mobility applications. This gap motivates research into dataset augmentation and synthetic data generation to extend recognition capabilities for new categories [145].

Two challenges are particularly critical in this context. First, the absence of the *Wheelchair* category in YOLOv8's pretrained weights restricts perception completeness in environments where wheelchairs are important dynamic agents. Second, inter-class ambiguity can reduce detection reliability: during preliminary evaluation, visually similar objects such as chairs and bicycles were often misclassified as wheelchairs. Such systematic errors increase the FP rate and undermine the dependability of the perception module in safety-critical indoor environments.

3.1.3 Contributions

This chapter contributes to the development of a robust perception backbone for safety-critical mobility through the following advances:

1. Introduction of the *Wheelchair* class into YOLOv8 by fine-tuning on a custom dataset, extending the detector beyond standard COCO categories.

2. Mitigation of systematic inter-class misclassification (e.g., wheelchair vs. chair or bicycle), thereby improving precision and reducing False Positive (FP) rates in cluttered indoor environments.

3. Establishment of a domain-adapted object detection framework that supports downstream explainability, monitoring, and uncertainty quantification mechanisms in later chapters.

3.2 Methodology

This section provides a detailed overview of the YOLOv8 object detection framework, its architectural components, and the rationale for its selection as the perception backbone in this work. The evolution of the YOLO family is first introduced, followed by an in-depth discussion of the YOLOv8 structure and its model variants. Finally, the concept of fine-tuning is defined in the context of adapting pretrained models to domain-specific requirements, such as wheelchair detection in indoor mobility environments.

3.2.1 Evolution of YOLO Models

The *You Only Look Once* (YOLO) family of models, introduced by Redmon et al. in 2015 [41], redefined object detection by framing it as a single regression task. Unlike traditional multi-stage pipelines that relied on region proposals followed by classification, YOLO directly predicts bounding boxes and class probabilities from an entire image in a single forward pass. This design enabled real-time detection with significant improvements in inference speed.

Subsequent versions introduced progressive refinements. YOLOv2 added batch normalization, multi-scale training, and anchor box mechanisms to improve detection performance [146]. YOLOv3 incorporated residual connections and multi-label classification, strengthening accuracy on small objects [147]. YOLOv4 advanced further by integrating Cross Stage Partial (CSP) networks and sophisticated data augmentation strategies, including Mosaic augmentation [143]. YOLOv5, developed by Ultralytics, transitioned the framework to PyTorch, emphasizing modularity, accessibility, and integration with modern machine learning pipelines [148].

YOLOv8, released in 2023, represents a major step in this progression, integrating anchor-free prediction, an optimized CSP backbone, and an enhanced PANet-based neck for multi-scale feature aggregation [144]. These innovations address the limitations of earlier versions, positioning YOLOv8 as a state-of-the-art detector for real-time and safety-critical applications.

3.2.2 YOLOv8 Architecture

The YOLOv8 architecture is composed of three primary components: the backbone, the neck, and the detection head [144].

Backbone: YOLOv8 employs an optimized CSPDarknet-based backbone for hierarchical feature extraction. This backbone captures both low-level visual details and high-level semantic information, with efficient use of depthwise separable convolutions to reduce computational overhead.

Neck: The neck fuses multi-scale features using an enhanced Path Aggregation Network (PANet), facilitating the detection of objects across diverse scales. This component improves the flow of semantic information while reducing redundancy in feature maps.

Head: The head predicts bounding boxes, confidence scores, and class labels. YOLOv8 departs from the anchor-based approach of earlier models by adopting anchor-free bounding box prediction. This innovation simplifies training, reduces the number of hyperparameters, and improves adaptability to objects with varying aspect ratios.

YOLOv8 further incorporates several methodological improvements that enhance both training efficiency and generalization. Advanced data augmentation strategies, such as mosaic and mixup, expose the model to diverse spatial arrangements and object scales, thereby improving robustness to cluttered indoor scenes. For classification, YOLOv8 employs a focal loss function, which mitigates class imbalance and increases the detection accuracy of rare or occluded objects. Localization is refined through an IoU-based loss, ensuring more precise bounding box regression. Finally, mixed precision training is adopted to accelerate the training process and reduce GPU memory consumption, enabling the use of larger batch sizes without compromising accuracy [144].

Figure 3.1 illustrates the overall architecture of YOLOv8, highlighting its backbone, neck, and anchor-free head, as presented in [144].

3.2.3 YOLOv8 Model Variants

YOLOv8 is released in multiple variants of increasing complexity: nano (YOLOv8n), small (YOLOv8s), medium (YOLOv8m), large (YOLOv8l), and extra-large (YOLOv8x).

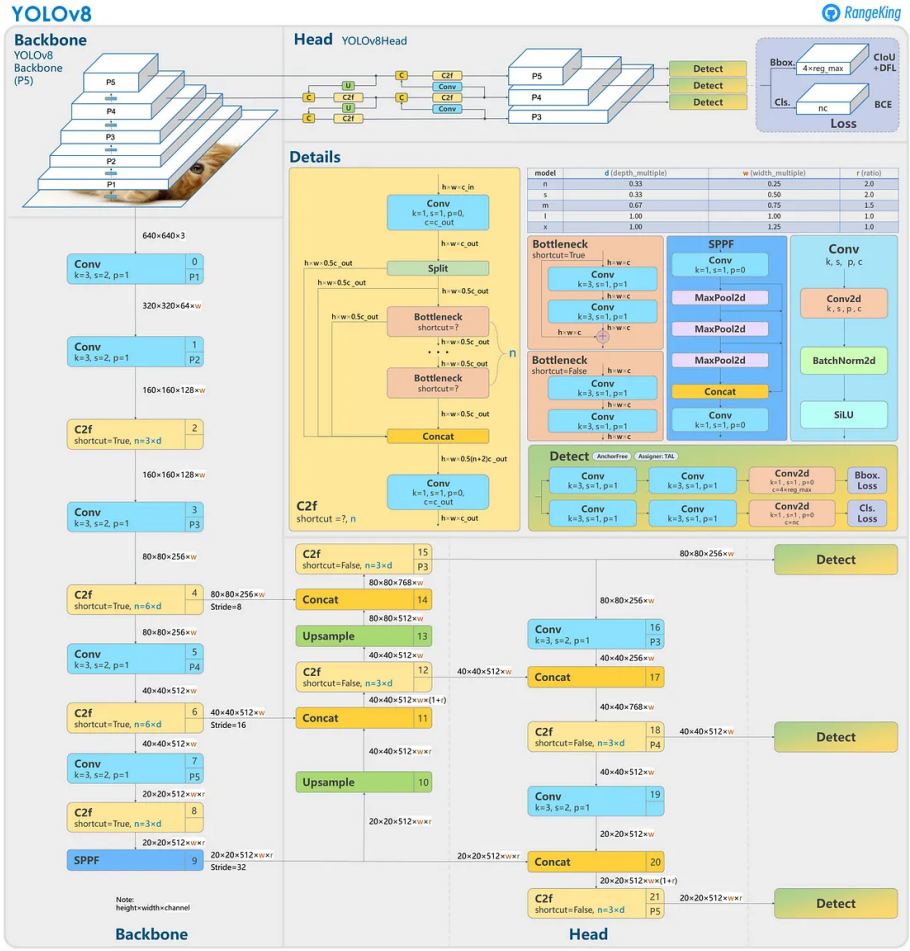


Figure 3.1: Model Structure of Yolov8

These models vary in the number of parameters, inference speed, and detection accuracy, allowing deployment across heterogeneous hardware platforms from edge devices to high-performance GPUs [144].

Table 3.1 summarizes the parameter sizes, accuracy (mAP@0.5), and inference times reported for the YOLOv8 family.

This scalability enables developers to trade off between computational efficiency and detection accuracy depending on deployment constraints.

3.2.4 Fine-Tuning for Domain Adaptation

Transfer learning has become the prevailing strategy to adapt pretrained DNNs to domain-specific tasks where annotated data is limited. Instead of training models from scratch, which requires large datasets and long convergence times, transfer learning leverages

Table 3.1: Performance comparison of YOLOv8 model variants on the COCO dataset, as reported in [144]. Inference times are measured on high-end GPUs (e.g., NVIDIA Tesla V100).

Model	Parameters (M)	mAP@0.5 (%)	GPU Inference (ms)
YOLOv8n	2.0	47.2	5.8
YOLOv8s	9.0	58.5	6.0
YOLOv8m	25.0	66.3	7.8
YOLOv8l	55.0	69.8	9.8
YOLOv8x	90.0	71.5	11.5

feature representations learned on large-scale datasets such as COCO [44]. In the context of object detection, this is typically realized through fine-tuning, where pretrained weights serve as initialization for a new task [46]. Early convolutional layers are often frozen to retain generic features such as edges and textures, while higher layers and detection heads are retrained to capture task-specific semantics. This approach has been shown to significantly reduce training time, improve convergence stability, and enhance performance under data-scarce conditions [45].

For the purposes of this work, fine-tuning was applied to adapt YOLOv8 to safety-critical categories not represented in standard benchmarks. Specifically, the *Wheelchair* class was introduced into the detector, extending perception capabilities to assistive mobility scenarios. In addition, targeted fine-tuning was employed to reduce systematic misclassifications between visually similar categories (e.g., wheelchairs, chairs, and bicycles), thereby lowering false positive rates and improving detection reliability in cluttered indoor environments. Through this process, a domain-adapted object detection backbone was established, forming the foundation for subsequent explainability and monitoring mechanisms developed in later chapters.

3.3 Experimental Validation

3.3.1 Datasets

To fine-tune the YOLOv8 model, a total of 8,813 images were compiled from public datasets, divided into 7,116 training images and 1,697 validation images. Two primary categories were defined for the task: *Person* and *Wheelchair*. Additional subsets containing visually similar objects (bicycle, chair, desk) were included to assess potential inter-class confusions. Beyond training and validation, a separate test set of 714 manually annotated

images was constructed from recordings at the Consiglio Nazionale delle Ricerche (CNR) Villa research facility in Genoa. This dataset was reserved exclusively for evaluating generalization under realistic deployment conditions. The employed datasets are described below.

3.3.1.1 Open Images Dataset V7

The Open Images Dataset V7 is a large-scale benchmark for object detection containing millions of bounding box annotations across more than 600 categories [149]. For this work, a subset of 7,678 images was extracted, focusing on the following categories:

Wheelchair Images containing wheelchairs or persons were collected from Open Images [150]. This subset formed the basis for introducing the *Wheelchair* category, which is absent in the pretrained YOLOv8 model. Examples are shown in Figure 3.2. The top row illustrates individual wheelchair instances, while the bottom row shows multi-object scenes with both wheelchairs and persons.



Figure 3.2: Sample images from the Open Images wheelchair subset.

Bicycle, Chair, and Desk Additional subsets of bicycles, chairs, and desks were also collected [151, 152]. In these images, only persons were annotated, while the furniture

and bicycle categories remained unlabeled. Although not directly used for training the *Wheelchair* class, these subsets (1,859 images) were included to analyze inter-class confusions, as wheelchairs were frequently mistaken for bicycles or chairs during preliminary experiments. Examples are shown in Figure 3.3.



Figure 3.3: Examples from the Open Images Subsets. Top: chairs and desks. Bottom: bicycles.

3.3.1.2 Roboflow Wheelchair Dataset

To increase the diversity of wheelchair instances, the Roboflow Wheelchair Dataset [153] was incorporated. This dataset contains 965 annotated images of wheelchairs, captured from a wide range of viewpoints and conditions. Its inclusion enhanced the robustness of the fine-tuned model to variations not present in Open Images. An example is shown in Figure 3.4.

3.3.1.3 Penn-Fudan Pedestrian Dataset

The Penn-Fudan Database for Pedestrian Detection and Segmentation contains 170 images with 345 labeled *Person* instances [154]. The dataset was collected in campus and urban street environments, with pedestrian heights ranging from 180 to 390 pixels. All



Figure 3.4: Example from the Roboflow Wheelchair Dataset, showing diverse wheelchair instances.

pedestrians are upright, and each image contains at least one labeled *Person*. This dataset was used to strengthen the *Person* category and to evaluate detection under real-world conditions such as occlusions and variable crowd densities. Examples are shown in Figure 3.5.

3.3.1.4 CNR Villa Dataset

A custom dataset was collected at the CNR Villa research facility in Genoa to serve as an independent benchmark for generalization. A set of indoor videos was recorded, from which 714 frames were extracted. All frames were manually annotated, yielding 3,149 labeled instances across the *Person* and *Wheelchair* categories.

The CNR Villa dataset was not used for training or validation. Instead, it was reserved exclusively as a dedicated test set to evaluate the generalization of the fine-tuned YOLOv8 model under realistic deployment conditions. Representative frames are shown in Figure 3.6.

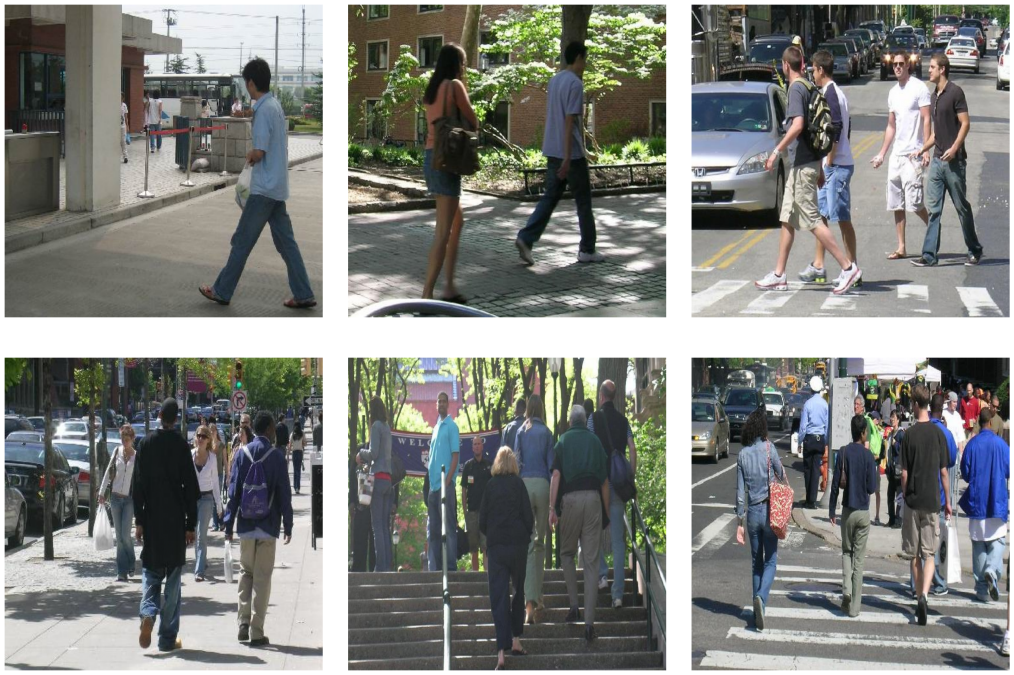


Figure 3.5: Examples from the Penn-Fudan Pedestrian Dataset

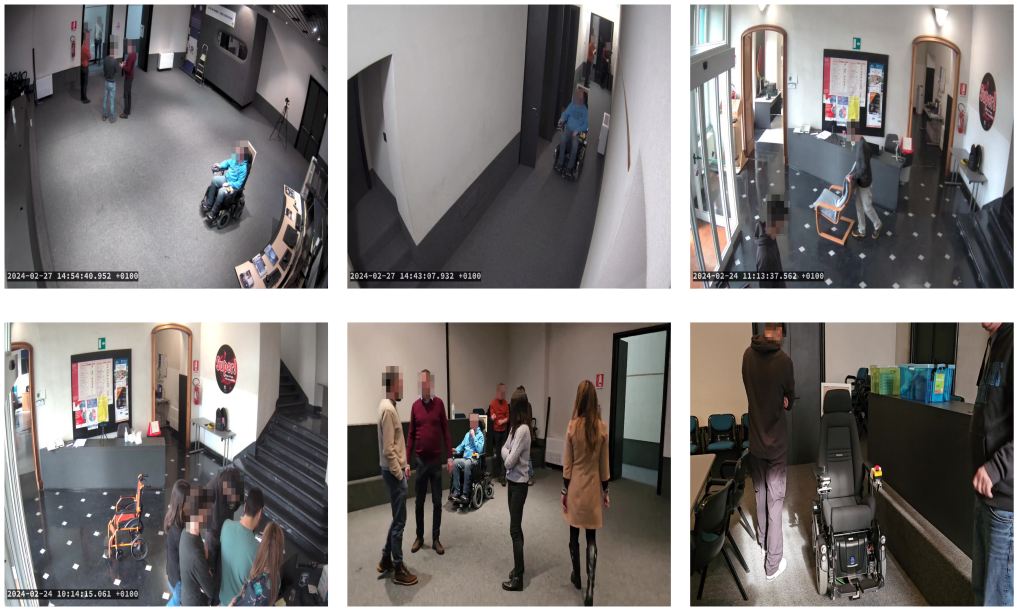


Figure 3.6: Examples from the CNR Villa Dataset. Frames were extracted from indoor video recordings and manually annotated for the *Person* and *Wheelchair* categories.

3.3.2 Training Configuration

The fine-tuning experiments were performed using the `YOLOv8s` variant as the base architecture, initialized with pretrained weights from the COCO dataset. An input resolution of 640×640 pixels was adopted, following the standard configuration in YOLO models that balances detection accuracy with computational efficiency. A batch size of 8 was chosen to accommodate GPU memory limitations while maintaining stable gradient updates.

Training was conducted for 150 epochs, which provided sufficient iterations for convergence while mitigating the risk of overfitting. To preserve the generic low-level features (e.g., edges, textures) learned from COCO pretraining, the first ten layers of the backbone were frozen. This strategy allowed the higher-level layers and detection head to specialize on the newly introduced *Wheelchair* category and adapt to the custom dataset distribution. Such a transfer learning approach is particularly effective when the target dataset is smaller in scale, as it accelerates convergence and improves generalization.

To further optimize training, image caching on disk was enabled to reduce input–output overhead and accelerate data loading. All experiments were executed on a dedicated GPU, ensuring consistent computational performance and reproducibility across training runs.

3.3.3 Evaluation Metrics and Performance Curves

To evaluate object detection models, a set of standard performance metrics and threshold-dependent curves are employed. These metrics quantify both the correctness of individual detections and the overall balance between False Positives (FPs) and False Negatives (FNs).

Intersection over Union (IoU). A predicted bounding box is considered a correct detection if it sufficiently overlaps with the ground-truth box. This overlap is quantified by the Intersection over Union (IoU), defined as

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \tag{3.1}$$

An illustration of IoU is provided in Figure 3.7.

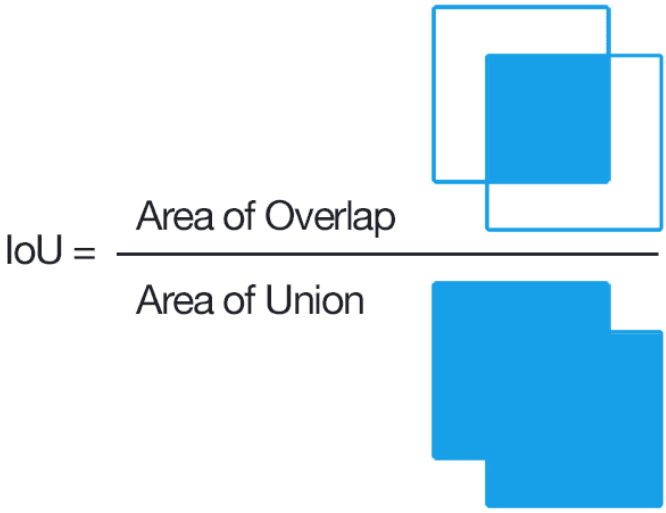


Figure 3.7: Illustration of Intersection over Union (IoU).

Precision. Given an IoU threshold (commonly 0.5), detections are classified as True Positive (TP) and FP. Precision is defined as

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{3.2}$$

Precision quantifies the reliability of detections by measuring the proportion of correctly identified objects among all predicted objects. High precision indicates few false alarms.

Recall. Similarly, recall is defined using TP and FN as

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3.3}$$

Recall measures completeness by quantifying the proportion of correctly detected objects relative to all ground-truth objects. High recall indicates that few objects were missed.

F1-score. The F1-score provides a single metric that balances precision and recall. It is defined as the harmonic mean of the two:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.4}$$

The F1-score is particularly useful when precision and recall exhibit trade-offs, as it emphasizes their joint optimization rather than favoring one over the other.

Average Precision (AP) and mean Average Precision (mAP). Average Precision (AP) is computed as the area under the precision–recall (PR) curve for a single class. The mean Average Precision (mAP) averages AP across classes. Two commonly reported values are:

mAP@0.5: AP evaluated at a single IoU threshold of 0.5, a relatively lenient criterion.

mAP@0.5:0.95: AP averaged over multiple IoU thresholds from 0.5 to 0.95 (step 0.05), as defined in the COCO benchmark. This provides a stricter measure of localization accuracy.

Performance Curves. Beyond scalar metrics, several confidence-dependent curves are used for deeper analysis:

Precision–Recall (PR) curve: shows the trade-off between precision and recall across thresholds.

F1–confidence curve: highlights the confidence threshold that maximizes the F1-score, the harmonic mean of precision and recall.

Precision–confidence and Recall–confidence curves: illustrate how each metric evolves as the confidence threshold is varied.

These curves provide insight into threshold selection and class-specific robustness, complementing the aggregate performance reported by mAP.

3.3.4 Results

The fine-tuned YOLOv8s model was evaluated on two benchmarks: the validation set derived from public datasets (1,697 images, 13,703 instances) and the CNR Villa dataset (714 images, 3,149 instances). The evaluation focused on the two target categories, *Person* and *Wheelchair*, while additional unlabeled distractor objects (bicycles, chairs, desks) were included to assess potential inter-class confusion.

3.3.4.1 Quantitative Results on Validation Set

Table 3.2 summarizes the performance metrics, including precision, recall, and mean Average Precision (mAP) at IoU thresholds of 0.5 and 0.5:0.95.

The results reveal a clear contrast between the two categories. *Wheelchair* detection achieved near-perfect recall (0.992) and very high precision (0.914), confirming the effectiveness of fine-tuning with targeted wheelchair datasets. In comparison, *Person* detection achieved reasonable precision (0.769) but lower recall (0.659), reflecting the greater variability, occlusions, and visual clutter that complicate pedestrian recognition in indoor environments.

On average, the model reached an mAP@0.5 of 0.846 and an mAP@0.5:0.95 of 0.656, demonstrating robust wheelchair recognition and acceptable generalization to persons.

Table 3.2: Validation performance of the fine-tuned YOLOv8s model. Metrics are reported per class and averaged across categories.

Class	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
Person	0.769	0.659	0.702	0.460
Wheelchair	0.914	0.992	0.990	0.852
All	0.842	0.825	0.846	0.656

3.3.4.2 Confusion Matrix Analysis

The normalized confusion matrix in Figure 3.8 highlights class-specific reliability. *Wheelchair*s were correctly identified in 99% of cases, while persons exhibited higher confusion with the background. These errors explain the recall gap observed in Table 3.2, indicating that failures in *Person* detection were predominantly FNs rather than misclassification as *Wheelchair*s.

3.3.4.3 Precision–Recall and Confidence Curves

Figures 3.9–3.12 provide further insight into threshold-dependent performance.

The F1-confidence curve (Figure 3.9) shows that the maximum F1-score of 0.83 occurs at a confidence threshold of 0.345, representing the optimal balance between precision and recall.

The precision-confidence curve (Figure 3.10) indicates that *Wheelchair* detection maintains high precision across all thresholds, whereas *Person* precision is more variable, particularly at lower confidence levels.

The recall-confidence curve (Figure 3.11) shows that wheelchair recall stays close to 1.0 across thresholds, while *Person* recall declines rapidly as the confidence threshold increases.

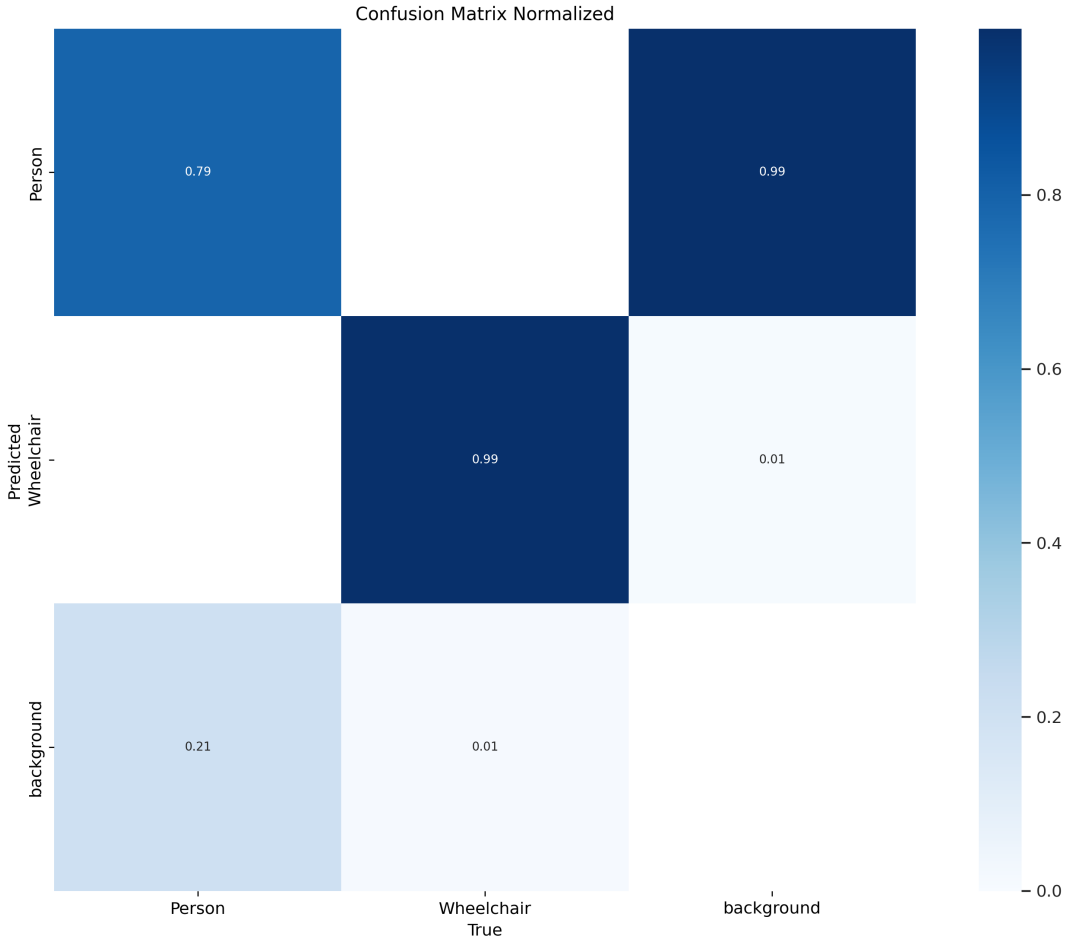


Figure 3.8: Normalized confusion matrix for the validation set showing *Person* and *Wheelchair* categories.

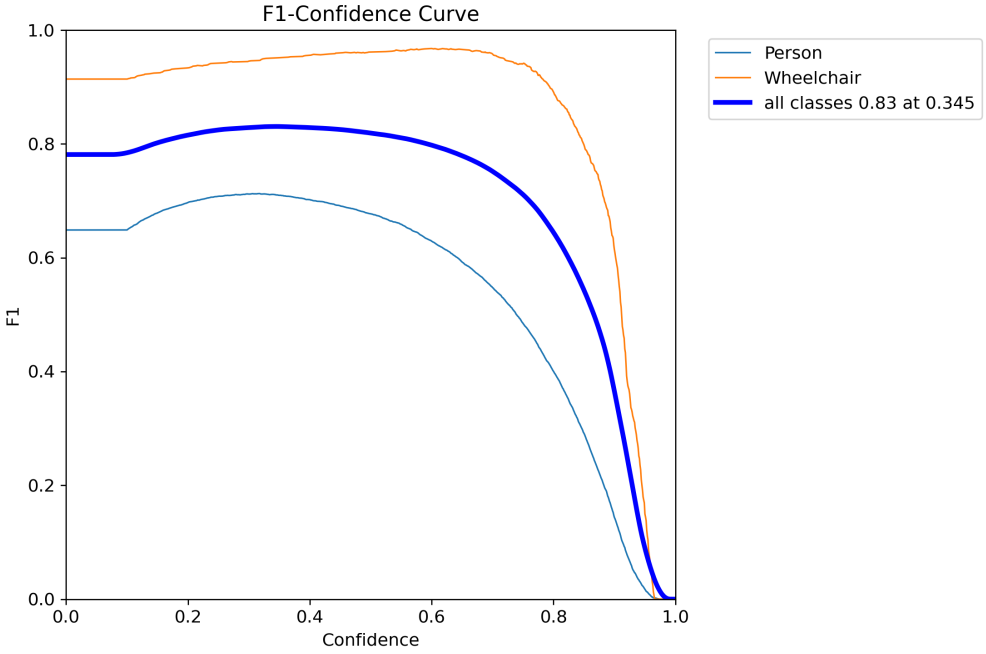


Figure 3.9: F1-confidence curve on the validation set.

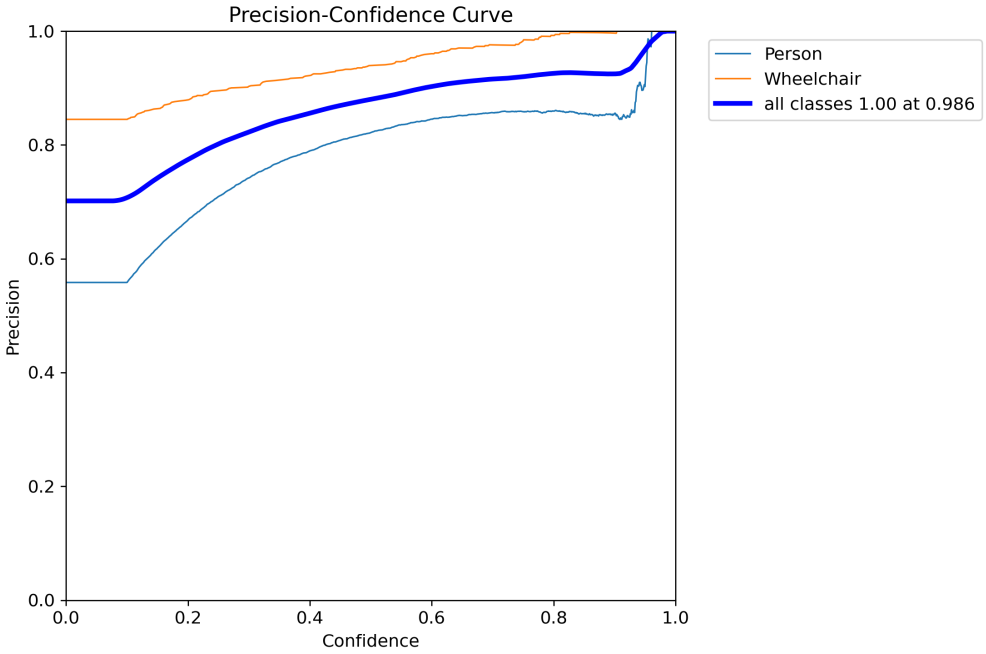


Figure 3.10: Precision-confidence curve on the validation set.

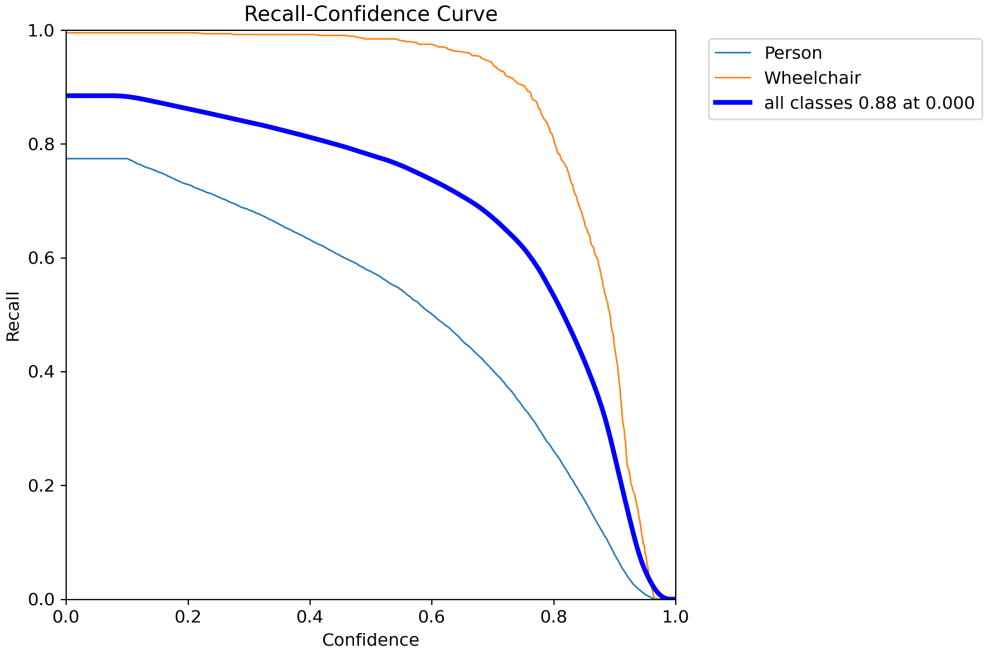


Figure 3.11: Recall-confidence curve on the validation set.

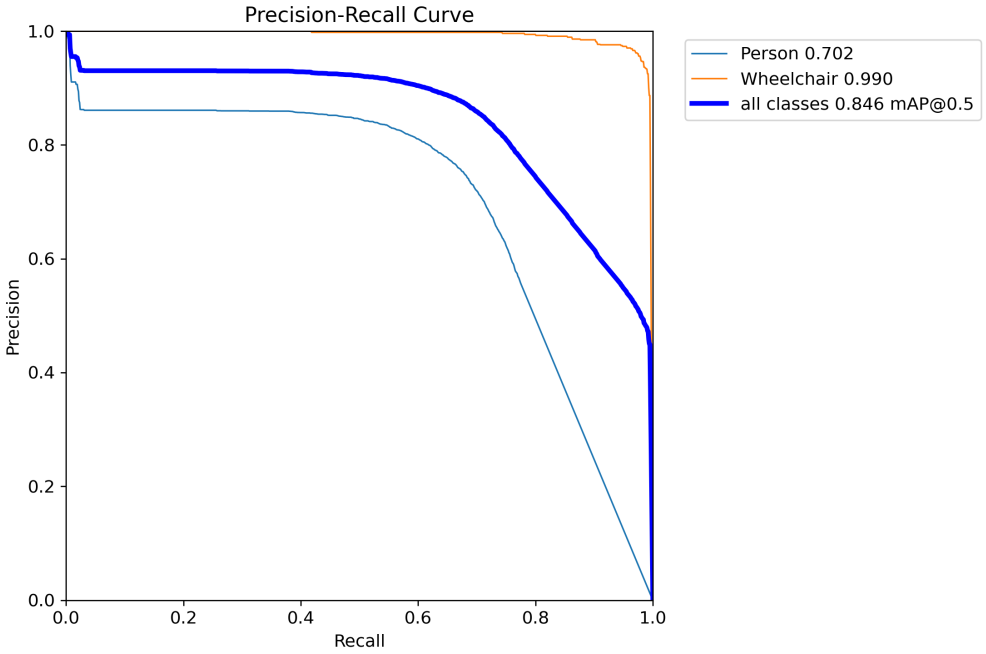


Figure 3.12: Precision–Recall curve on the validation set.

Finally, the precision–recall curve (Figure 3.12) emphasizes the disparity between the two classes. *Wheelchair*s attained an mAP@0.5 of 0.990, whereas *Person*s achieved 0.702. This highlights both the success of fine-tuning for the *Wheelchair* category and the inherent difficulty of robust *Person* detection in complex indoor environments.

3.3.4.4 Results on the CNR Villa Dataset

To assess generalization, the fine-tuned YOLOv8s model was evaluated on the independent CNR Villa dataset, consisting of 714 annotated frames and 3,149 labeled instances.

Table 3.3 presents the quantitative results. The model achieved an overall precision of 0.972, recall of 0.963, and mAP@0.5 of 0.989, with a strong mAP@0.5:0.95 of 0.858. Both classes, *Person* and *Wheelchair*, achieved comparable results, confirming the robustness of the fine-tuning strategy in practical scenarios.

Table 3.3: Performance of the fine-tuned YOLOv8s model on the CNR Villa dataset. Results are reported per class and averaged across categories.

Class	**Precision**	**Recall**	**mAP@0.5**	**mAP@0.5:0.95**
Person	0.974	0.958	0.989	0.877
Wheelchair	0.970	0.967	0.989	0.839
All	0.972	0.963	0.989	0.858

The normalized confusion matrix in Figure 3.13 confirms these findings. Both classes were detected with very high reliability: *Wheelchair*s were correctly identified in 99% of cases, while *Person*s exceeded 95% recall despite occasional misclassification as background.

Confidence-dependent analyses further support the robustness of the model. The F1-confidence curve (Figure 3.14) indicates that the maximum F1-score of 0.97 is achieved at a confidence threshold of 0.533. The precision–recall curve (Figure 3.15) shows nearly identical behavior for both classes, with an mAP@0.5 of 0.989, reflecting strong balance between precision and recall in practical conditions.

Finally, qualitative results were obtained by overlaying detection outputs on representative frames from the dataset. Figure 3.16 illustrates robust detection of both *Person* and *Wheelchair* across varied crowd densities and viewpoints, providing visual evidence of the model's reliability under real-world indoor conditions.

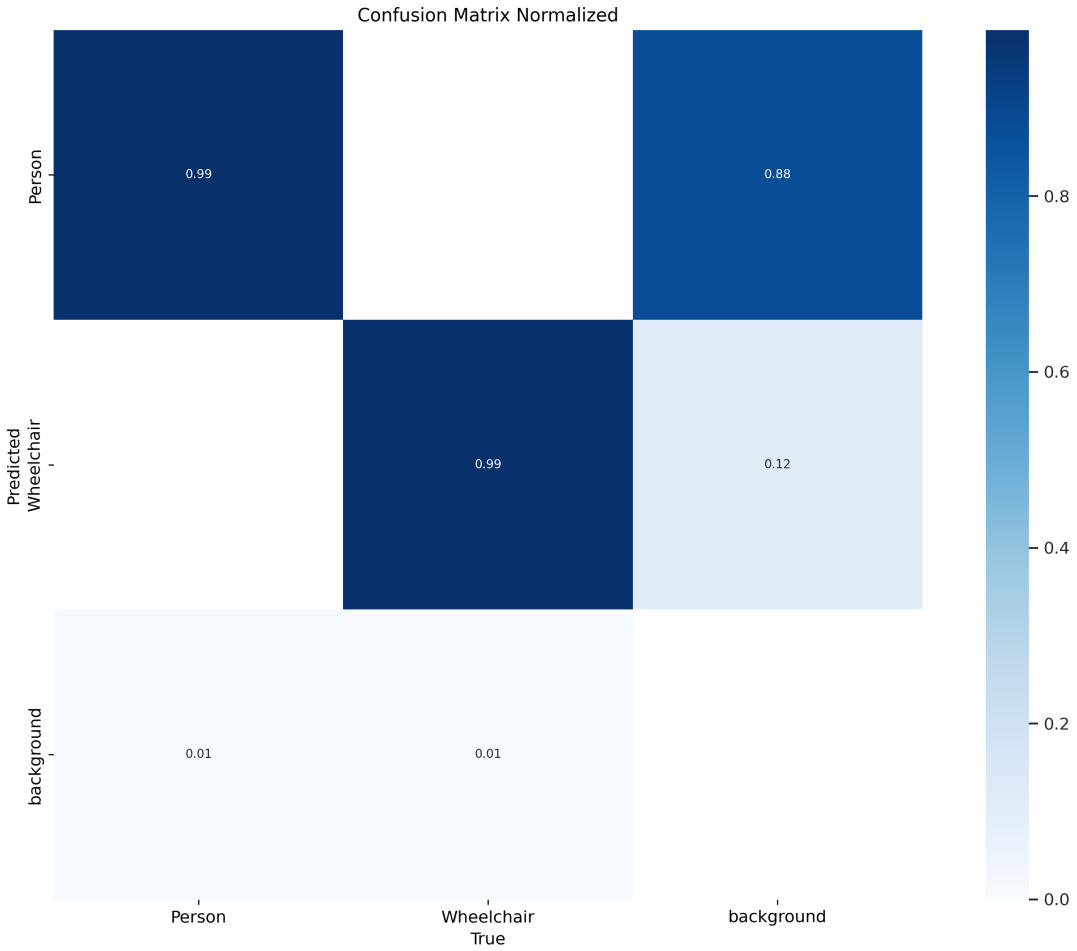


Figure 3.13: Normalized confusion matrix on the CNR Villa dataset for the *Person* and *Wheelchair* categories.

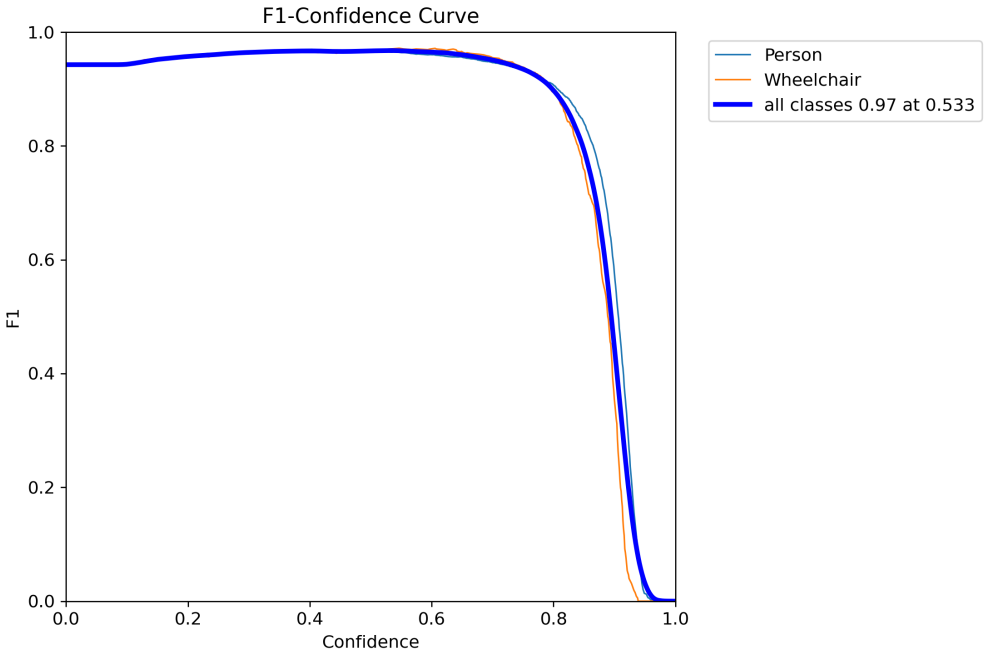


Figure 3.14: F1-confidence curve on the CNR Villa dataset

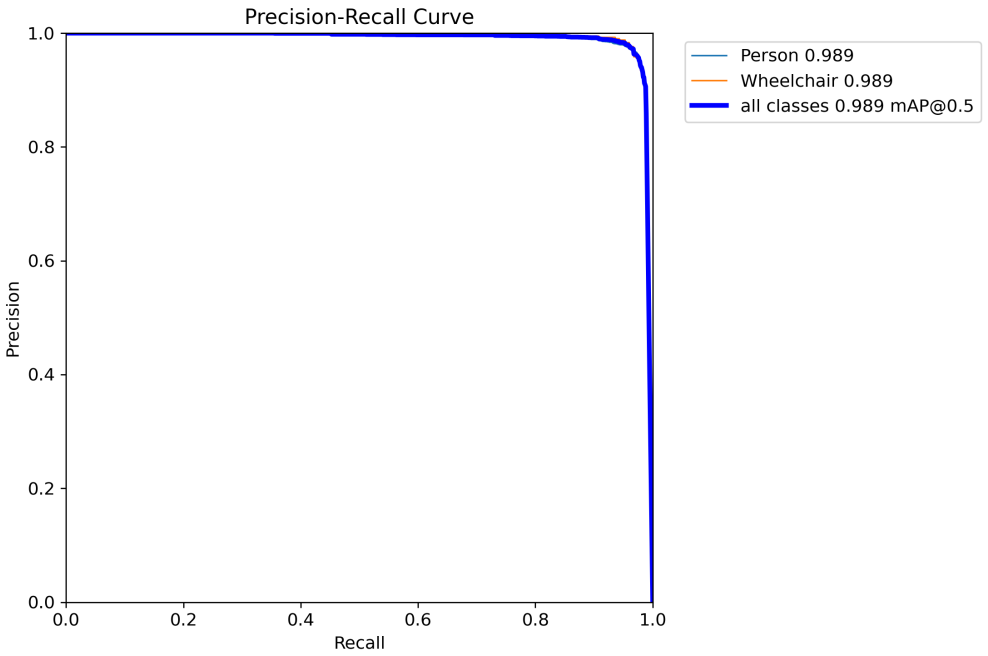


Figure 3.15: Precision–Recall curve on the validation set for the *Person* and *Wheelchair* categories.



Figure 3.16: Qualitative detection results on the CNR Villa dataset.

3.4 Conclusions

Object detection is a cornerstone of perception in autonomous and assistive mobility systems, enabling the reliable identification and localization of critical objects in complex environments. This chapter presented the fine-tuning and evaluation of the YOLOv8 object detection framework for safety-critical indoor mobility scenarios. The motivation stemmed from the need to extend perception capabilities beyond generic pretrained datasets by introducing the *Wheelchair* category and addressing systematic misclassifications with visually similar objects such as bicycles and chairs. Through this adaptation, a robust perception backbone was established for deployment in realistic indoor environments.

The fine-tuning process leveraged transfer learning, freezing low-level backbone layers while retraining higher-level components and the detection head. A diverse combination of datasets supported this process: Open Images subsets, the Roboflow Wheelchair Dataset, and the Penn-Fudan Pedestrian Dataset provided complementary training and validation data, while the CNR Villa dataset was reserved as an independent test set to evaluate generalization under deployment-like conditions. This systematic design ensured both domain adaptation and rigorous benchmarking.

The experimental evaluation confirmed the effectiveness of the approach. On the validation set, *Wheelchair* detection achieved near-perfect recall (0.992) and high precision (0.914), with an mAP@0.5 of 0.990, while person detection exhibited lower recall due to greater intra-class variability and frequent occlusions. On the independent CNR Villa dataset, the model demonstrated excellent generalization, with both categories reaching an overall mAP@0.5 of 0.989 and consistently balanced precision–recall trade-offs. Qualitative analyses further illustrated robust operation across varied viewpoints and crowd densities.

Beyond raw performance, this chapter contributes a reliable foundation for subsequent work. By ensuring dependable detection of *Person*s and *Wheelchair*s, the perception module enables higher-level safety functions. In particular, it provides the necessary backbone for explainability analysis and uncertainty quantification, which will be developed and explored in the following chapters.

CHAPTER 4

eXplainable Checker of Video Analytics Performance in Indoor Smart Mobility

4.1 Introduction

Building directly on the perception backbone introduced in Chapter 3, this chapter embeds explainability into video analytics by introducing a novel rule-based Operational design domain (ODD) verification layer. The proposed Checker complements raw object detection with explicit, auditable decision boundaries, enabling interpretable monitoring of prediction confidence in indoor smart mobility scenarios.

4.1.1 Background

Autonomous Driving Systems (ADS) and Advanced Driver-Assistance Systems (ADAS) are leading the way in innovation in mobility technologies, offering transformative solutions spanning a broad range of domains—from Autonomous Vehicles (AV) to robotics and assistive devices. In the last decade, these systems have seen tremendous advances, powered by breakthroughs in computer vision and data-driven methodologies [155]. With this advancement, ensuring the safe deployment of autonomous systems is of paramount importance [156]. A crucial element of this effort is understanding *"how a system fails"*, that is, systematically discovering its limitations [157].

ADS and ADAS typically face extremely complex scenarios, such as urban roads, where the conditions change with dynamic weather and unpredictable actors like pedestrians. While these challenges are well-recognized, this chapter narrows the focus to indoor smart mobility environments (e.g., stations, airports, hospitals), where dynamic pedestrian flow, occlusions, and constrained spaces pose unique perception and safety verification challenges. These unique challenges motivate a deeper examination of safety verification mechanisms, which are discussed in the following subsections.

4.1.2 Key Issues in Autonomous Driving Systems (ADS) Safety

Safety in ADS comprises two critical aspects: functional safety and Safety Of The Intended Functionality (SOTIF) [158, 159]. Functional safety pertains to system failures of hardware or software, which are typically covered by existing safety standards. In contrast, SOTIF, according to the definition of ISO/PAS 21448, puts a focus on risks associated with a system's intended functionality, such as perception limitations, decision-making limitations, and user errors. Although functional safety primarily addresses issues related to system failures, SOTIF highlights difficulties that emerge even during proper system operation.

Ensuring SOTIF in ADS is especially difficult owing to both the environmental variability and complexity in real-world scenarios. ADS systems are built upon Artificial Intelligence (AI) and Machine Learning (ML) models as key elements. These models typically assume that data distributions must remain unchanged between training and deployment, but in practice, this assumption can hardly be met due to the domain shifts. For instance, the rapidly changing light condition or complex traffic environment that has never been trained before would easily make the system behave unpredictably and thus compromise the safety of the system [160]. Hence, these are critical systems for ensuring both regulatory compliance and safety protection. For this reason, Operational design domain (ODD) monitoring becomes a cornerstone of guaranteeing safe and reliable operation of ADS.

4.1.3 Open Challenges

One of the major open challenges is to define rigorous standards for ADS, in particular, when considering the growing complexity and opacity of AI/ML models. Although a number of existing standards like ISO 26262 on functional safety and ISO/PAS 21448 on SOTIF provide a good starting point, they are not armed sufficiently to deal with the unique challenges raised by highly complex ADS. Regulatory frameworks often focus on performance standards rather than the specifics of technology, guiding the focus toward the end product rather than the methods used. However, this approach causes concern in cases of untrained scenarios because AI models may produce unpredictable outcomes. In addition, in adaptive systems that evolve over time based on experiential learning, it becomes difficult to set clear and enforceable performance standards.

The SAE J3016 framework defines multiple levels of automation, and Level 4 and above ADSs have advanced components and features [161]. Central to such systems is

the concept of an ODD, which delineates the environmental, geographical, temporal, and roadway parameters that specify the safe operating domain of an ADS. When operating within these limits, the system should fulfill its intended functionalities; exceeding these limits requires moving the system to a safe state. Nevertheless, providing real-time verification of these limits increases the complexity even further. A very important regulatory challenge is to answer how the current operational domain—the environment where the ADS or ADAS operates—is compatible with the defined ODD it has been designed and validated for. The activity usually referred to as ODD monitoring, is the continual checking of real-time environmental conditions against pre-defined specifications. Deviations from the intended ODD can lead to hazardous behaviors, hence, these are critical systems for ensuring both regulatory compliance and safety protection. To address these gaps, eXplainable Artificial Intelligence (XAI) has emerged as a promising direction for enhancing the transparency and reliability of ADS safety mechanisms.

4.1.4 Explainable Artificial Intelligence (XAI) for Safe Autonomous Driving Systems (ADS)

Explainability in AI-enabled systems, including ADS and ADAS, has become an active area of research, driven by the 'black box' nature of AI models and the need for transparency in safety-critical scenarios. A. Kuznietsov et al. [162] categorize explainability in ADS into several approaches, including interpretable design, interpretable surrogate models, interpretable monitoring, auxiliary explanations, and interpretable validation. These approaches collectively aim to enhance the transparency and trustworthiness of AI-based autonomous driving systems by providing insights into their decision-making processes.

In general, interpretability refers to the ability of an algorithm or system to provide intelligible explanations for its outputs or behavior. For example, an interpretable-by-design model explicitly reveals causal relationships between inputs and outputs [163], while surrogate models approximate the behavior of black-box algorithms to provide understandable explanations [164]. Interpretable monitoring systems, which are particularly relevant to this work, use interpretable-by-design models to verify decision-making outputs, ensuring safer deployment of autonomous vehicles. Similarly, auxiliary explanations and safety validation methods generate additional information or simulate adversarial behaviors to assess and enhance the safety of ADS. In this chapter, we build on the concept of interpretable monitoring systems by integrating rule-based explanations with visual feature analysis for real-time ODD safety verification.

4.1.5 Contributions

This chapter presents a novel framework for ODD monitoring, introducing a safety verification layer through an XAI methodology integrated with visual feature analysis of images. The main contributions are:

1. A novel framework for ODD monitoring through interpretable rule-based verification.
2. Integration of visual feature analysis with XAI methodologies to formalize ODD definitions.
3. Robustness enhancement via systematic image perturbations, addressing environmental and visual variability.
4. Real-time deviation detection and alerting mechanism, with preprocessing to improve input data quality.

4.2 Methodology

The proposed workflow for ODD monitoring within a real-time object detection system is presented in Figure 4.1. In order to add a safety verification layer, the architecture is designed as a dual-path pipeline. The main path performs the object detection task by scanning the input image to identify the target objects (Sect. 4.2.1). In parallel, the secondary pathway extracts important visual features from the same input (Sect. 4.2.2), generating a secondary dataset where each image is described by its extracted features. These features are then analyzed by an ODD Checker, which evaluates their conformity to predefined ODD boundaries. As long as the extracted features are within the predefined boundaries of the ODD, the object detector output is classified as *Certain*. However, if these features fall outside the ODD limits, the output is classified as an *Uncertain* prediction, signaling a potential deviation from the defined operational conditions.

The content of this chapter has been published in the following paper:

A. Samandari, J. Motta, V. Orani, A. Carlevaro, and M. Mongelli, "eXplainable Checker of Video Analytics Performance in Indoor Smart Mobility," in *Intelligent Systems Conference (IntelliSys 2025)*, K. Arai (Ed.), Lecture Notes in Networks and Systems, vol. 1554, pp. 1–20, Springer, 2025. doi: 10.1007/978-3-031-99965-9_24.

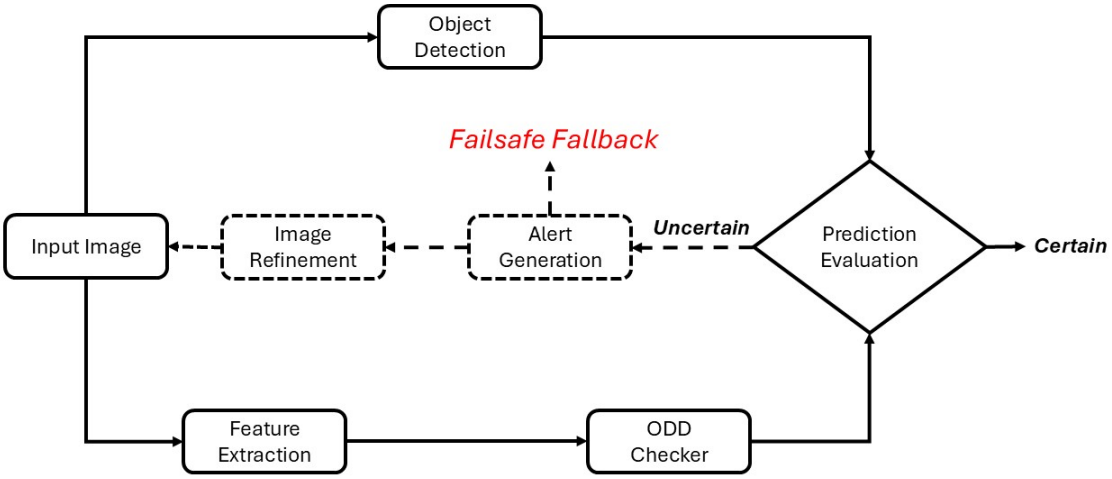


Figure 4.1: Real-time ODD Monitoring System Architecture

This section describes the procedure for developing the ODD Checker. A secondary dataset is first prepared by pairing extracted visual features with certainty labels. These labels (*Certain* or *Uncertain*) are assigned by evaluating object detection results against ground truth, as detailed in Sect. 4.2.3. This results in a fully labeled dataset that captures both visual characteristics and corresponding prediction certainty. Finally, an interpretable XAI classifier is trained on this data to learn decision rules that define the operational boundaries of the ODD (Sect.4.2.4). These rules are integrated into a lightweight module that performs real-time certainty assessment during system operation (Sect.4.2.5).

To extend this architecture toward autonomous correction, the workflow also outlines two envisioned components: an alert generation module that triggers a failsafe fallback and an image refinement stage aimed at improving prediction reliability. These auxiliary components represent possible extensions aimed at enhancing system resilience.

4.2.1 Object Detection

The object detector used in this chapter is the YOLOv8 model, introduced in Chapter 3. For the purposes of ODD monitoring, it was specifically configured to detect two safety-critical categories — Person and Wheelchair — due to their relevance in indoor smart mobility applications.

4.2.2 Feature Extraction

In parallel to object detection, visual features are extracted from each input image to enable rule-based ODD monitoring. In this study, features were extracted using the Athec library [165, 166], which quantifies aesthetic and structural image properties relevant for visual perception. For clarity, the features can be grouped into the following categories:

Color and Intensity Features: Derived from multiple color spaces (RGB, HSV, HSL, Lab, XYZ), these features capture luminance, chromaticity, and perceptual color differences that affect detection robustness.

Grayscale and Contrast Metrics: Include grayscale intensity statistics, global/local contrast, and colorfulness indices, providing cues on lighting conditions, visual sharpness, and scene richness.

Structural Features: Edge complexity computed using Canny edge detection captures the amount of fine detail and geometric patterns, which are critical under cluttered or low-texture conditions.

Saliency Features: Computed with spectral residual and fine-grained methods, these highlight visually dominant regions, aiding the analysis of object visibility under occlusions or distractions.

Together, these feature groups offer a comprehensive view of the image's aesthetic and structural content, supporting the analysis of prediction certainty under varying visual conditions.

4.2.3 Validation

Validation in this study serves two critical purposes: during the boundary definition phase, it evaluates the object detection model's performance against ground truth annotations to generate labels for the secondary dataset. Subsequently, once the boundaries of the ODD are established, validation involves assessing the features of input images and comparing them to the predefined boundaries to ensure compliance.

During the boundary definition phase, each image in the dataset is assigned a label of either *Certain* or *Uncertain*, based on the detection certainty. The validation process considers two distinct scenarios, depending on the number of objects in the image:

Single Object: An image is labeled as *Certain* if the Intersection over Union (IoU) between the detected object and the ground truth object is at least 0.7.

Multiple Objects: An image is labeled as *Certain* if all the following conditions are satisfied:

Each detected object has an IoU of at least 0.7 with its corresponding ground truth object.

At least 60% of the ground truth objects are correctly detected ($T_d = 0.6$).

The proportion of False Positive (FP)s does not exceed 40% of the total ground truth objects.

To compute the allowable number of FPs (N_{FP}), the following formula is used:

$$N_{FP} = (1 - T_d) \times N_{GT} \tag{4.1}$$

where:

T_d is the detection threshold (e.g., $T_d = 0.6$).

N_{GT} is the total number of ground truth objects in the image.

For instance, for an image with $N_{GT} = 10$ ground truth objects, if 7 objects are correctly detected with an IoU of at least 0.7, and no more than 4 FPs ($N_{FP} = 4$) are identified, the detection satisfies the certainty criteria, and the image is labeled as *Certain*.

Images that do not meet the *Certain* criteria are labeled as *Uncertain*. These cases typically arise when a significant portion of the ground truth objects are either not detected or incorrectly detected, or when the proportion of FPs exceeds acceptable thresholds.

Once the ODD boundaries are defined, the validation process shifts to evaluating the features of incoming images in real-time. The extracted features of each input image are compared to the predefined ODD boundaries established during the boundary definition phase. This ensures that the system can detect deviations and identify cases where predictions may fall outside the safe operating domain.

Since this study focuses on analyzing two specific object categories, the secondary dataset is further divided into two primary categories: Person and Wheelchair. The criterion for assigning an image to a category is that it must contain at least one object of the corresponding class. This division enables detailed performance analysis by object class, ensuring a comprehensive assessment of the system across distinct object types. The

resulting labeled datasets serve as inputs for subsequent stages involving XAI classification and rule generation.

4.2.4 Feature Selection and Rule Extraction

Decision Trees (DTs) were selected due to their inherent interpretability, allowing the system to generate human-readable rules for ODD compliance. Once the labeled secondary datasets were obtained (as described in Sect. 4.2.3), separate DT classifiers were employed for the Person and Wheelchair categories to address two key objectives: feature selection and rule extraction. Each image in the datasets was characterized by a large set of visual features extracted using the Athec library. This high-dimensional feature space introduced significant computational overhead, especially for real-time applications. To address this, a Cumulative Feature Ranking (CFR) approach was integrated with the DT to rank and select the most relevant features. In each iteration, 40% of the dataset was randomly sampled, and a DT classifier was trained using optimal hyperparameters selected through grid search. Feature importances were computed and accumulated with weighted scores based on their ranks. After multiple iterations, a final cumulative ranking was generated, ensuring robust and consistent feature selection while mitigating overfitting and reducing computational complexity.

By incrementally narrowing the feature set to the most informative attributes, the CFR method substantially decreased computational demands, enabling efficient real-time processing without compromising detection performance. This streamlined feature selection is particularly critical for dynamic environments requiring rapid decision-making, such as object detection in autonomous systems.

After identifying the most important features through CFR, the datasets corresponding to each category were split into training and evaluation subsets: 60% of the data was allocated for training and rule extraction, while the remaining 40% was reserved for evaluating the system's ability to classify prediction certainty. During the training phase, the DTs learned to create rules that describe decision boundaries for *Certain* and *Uncertain* predictions. These extracted rules form the foundation of the ODD Checker, which enables interpretable certainty evaluation during deployment.

4.2.5 Real-Time Certainty Evaluation

After training, the ODD Checker becomes the core component for evaluating whether new inputs conform to the defined decision boundaries of the ODD. During deployment, the system operates in real time, analyzing each input using only the most relevant visual features identified in Sect. 4.2.4. These features represent the operational conditions under which the object detector is expected to perform reliably.

If the extracted features fall within the rule-based boundaries, the associated prediction is classified as *Certain*; otherwise, it is labeled as *Uncertain*. This mechanism allows for a lightweight and interpretable assessment of prediction certainty, without relying on the internal logic of the object detector.

4.3 Experimental validation

4.3.1 Object Detector: Dataset and Configuration

As detailed in Chapter 3, the YOLOv8 object detector was fine-tuned on publicly available datasets combining wheelchair and Person instances. The training corpus included subsets from the Open Images Dataset [149, 150, 151, 152] and the Roboflow Wheelchair Dataset [153], supplemented with bicycle and chair images to reduce inter-class confusions. The final training set contained 7,116 images, with 1,697 reserved for validation.

For efficiency and deployment feasibility, the YOLOv8s (small) variant was adopted. Transfer learning was applied by freezing the first ten backbone layers while fine-tuning the remaining layers on the task-specific dataset. Training was performed for 150 epochs with a batch size of 16.

4.3.2 ODD Checker Dataset

The datasets used to construct and evaluate the ODD Checker were drawn from two independent indoor sources: the CNR Villa research facility in Genoa, Italy, and Aitek SpA. These datasets formed the basis for the secondary dataset used in validation and for training the XAI classifier, ensuring that both controlled and naturally variable conditions were represented.

The CNR Villa dataset, previously described in Section 3.3.1.4, consists of 714 annotated images. For this work, the dataset was further extended through controlled perturbation schemes designed to emulate challenging real-world conditions that may

compromise object detection performance. Six perturbation types were applied: compression, contrast adjustment, brightness modification, blurriness, darkness, and circular darkness. Each perturbation type was implemented at three intensity levels, producing 18 perturbed variants per original image. Additionally, one random perturbation was generated by combining functions and intensities, yielding 19 variants per original image in total. An illustration of these perturbations is provided in Figure 4.2.

In contrast, the Aitek SpA dataset consisted of 265 images recorded under natural indoor conditions without controlled perturbations. These images inherently reflected diverse real-world challenges such as varying lighting, occlusions, and cluttered environments, and were therefore used in their original form. A representative sample is shown in Figure 4.3.

The combined dataset, consisting of both original and perturbed images, amounted to 14,545 samples across the two categories of interest (*Person* and *Wheelchair*). Each sample was validated and labeled according to prediction certainty, forming the foundation for reliable rule extraction in the ODD Checker. A summary of the dataset distribution is reported in Table 4.1.

Table 4.1: Validation dataset breakdown for Person and Wheelchair categories.

Category	**Certain Predictions**	**Uncertain Predictions**	**Total Images**
Person	8140	5770	13,910
Wheelchair	7090	2642	9732

4.3.3 Results

4.3.3.1 Feature Ranking and Importance

To identify the most impactful features, the cumulative feature ranking (CFR) process (Sect. 4.2.4) was applied. In each iteration, 40% of the dataset was sampled and feature importances were computed; this was repeated for 100 iterations to ensure robustness and stability of the ranks. The optimal number of features was determined by evaluating varying feature set sizes and analyzing their effects on the False Negative Rate (FNR) and False Positive Rate (FPR). As shown in Figure 4.4, selecting 20 features provided the best trade-off, minimizing FNR (particularly for the Person category) while maintaining low FPR for both Person and Wheelchair datasets. Figures 4.5 and 4.6 report the top-ranked features for each category.

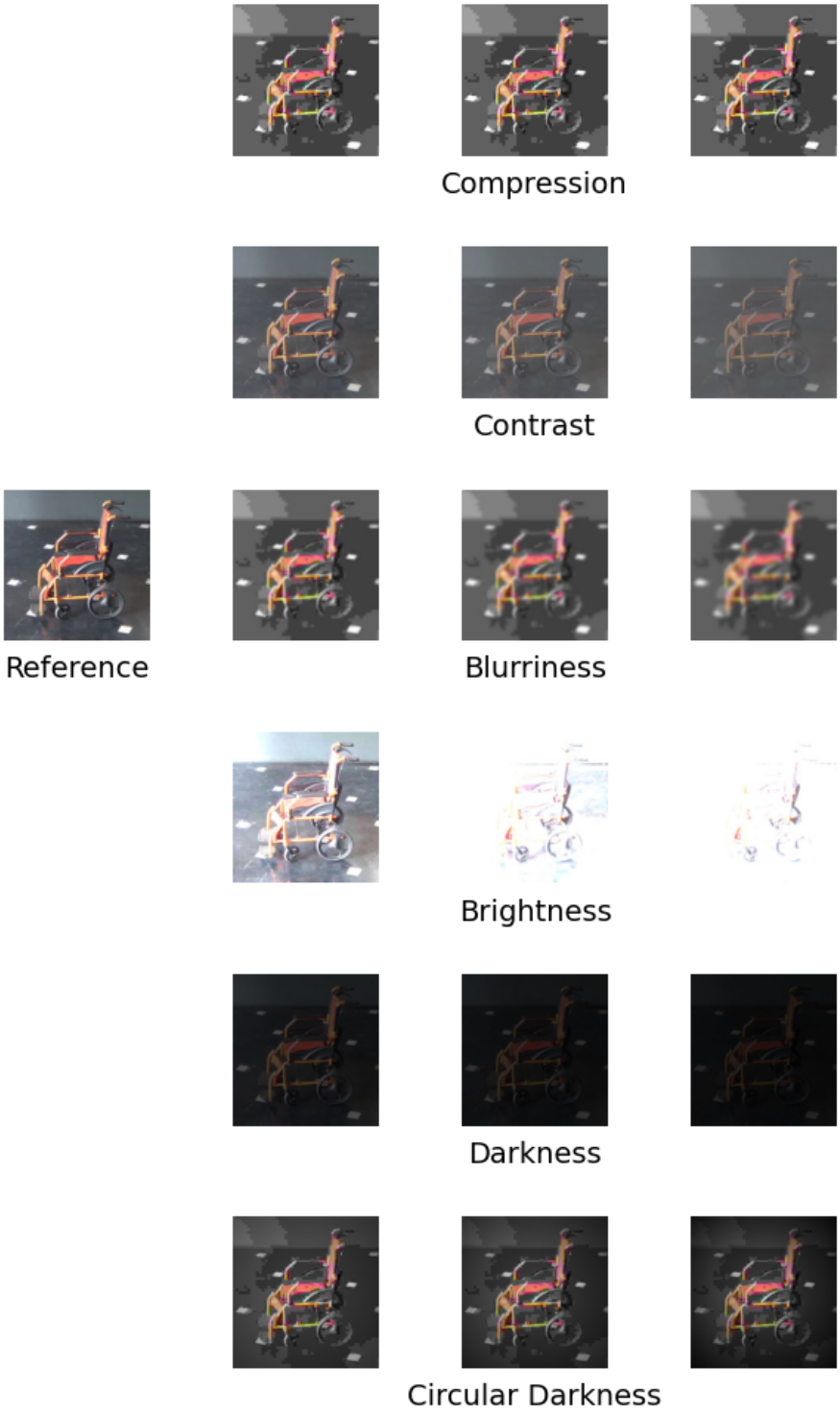


Figure 4.2: Example perturbations of a CNR Villa dataset sample at different intensity levels.

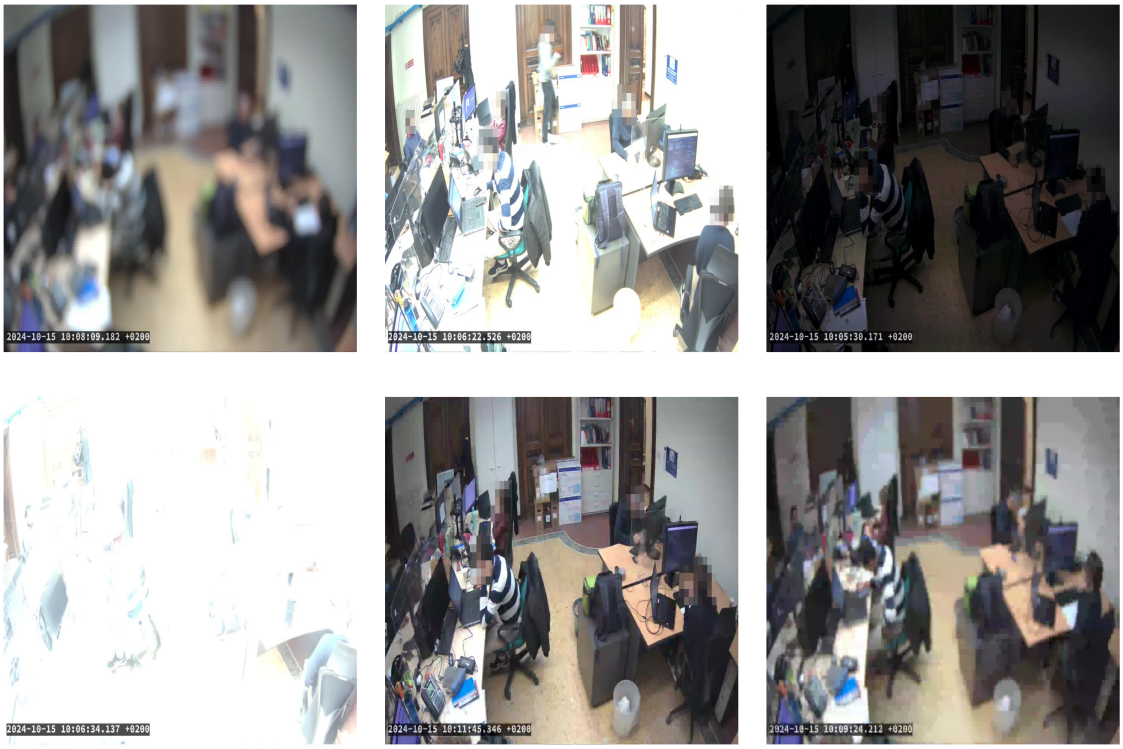


Figure 4.3: Representative examples from the Aitek SpA dataset.

Person. Features such as *HSV_H_mean* and *HSL_H_mean* emerged as the most critical, highlighting the importance of hue across HSV and HSL spaces. Red-channel statistics (*RGB_R_max*, *RGB_R_kurtosis*), saliency descriptors (*saliency_block_93*), and structural/contrast cues (*edge_density*, proportion of *white* regions) further strengthened discrimination between *Certain* and *Uncertain* cases.

Wheelchair. *HSL_H_mean*, *HSV_H_mean*, and *HSV_H_circular_mean* dominated the top ranks, confirming the centrality of hue-based attributes. Red-channel complexity (e.g., *RGB_R_entropy*) and luminance (*Lab_L_mean*) contributed to separating *Certain* from *Uncertain* predictions. Saliency blocks (e.g., *saliency_block_95*, *saliency_block_79*) captured contextual structure, while *edge_density* and *RGB_B_skew* reflected texture and chromatic variability.

Shared indicators. Despite category-specific variations, both datasets converged on consistent indicators of prediction reliability. Hue-based attributes (*HSV_H_mean*, *HSL_H_mean*, *HSV_H_circular_mean*), luminance metrics (*Lab_L_mean*), and struc-

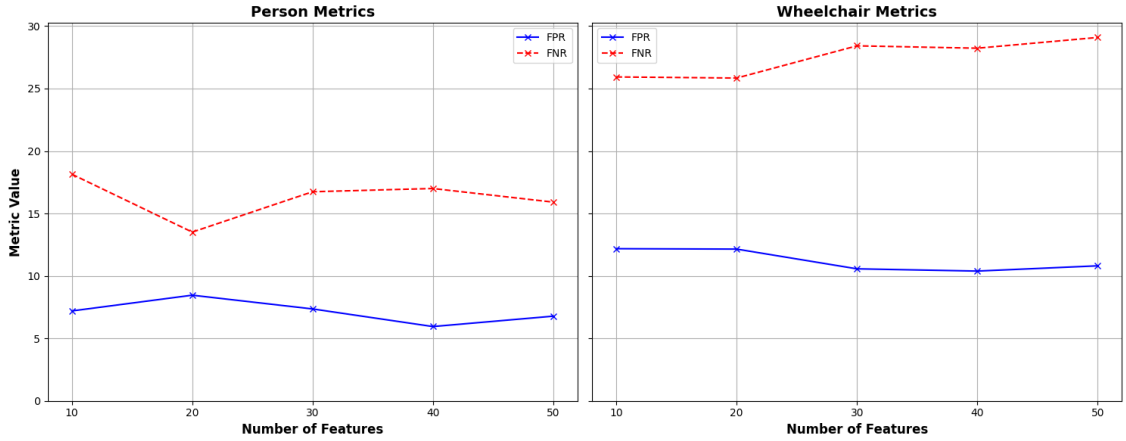


Figure 4.4: Trade-off analysis of False Negative Rate (FNR) and False Positive Rate (FPR) for different feature set sizes in the Person and Wheelchair categories. Optimal performance is achieved at 20 features.

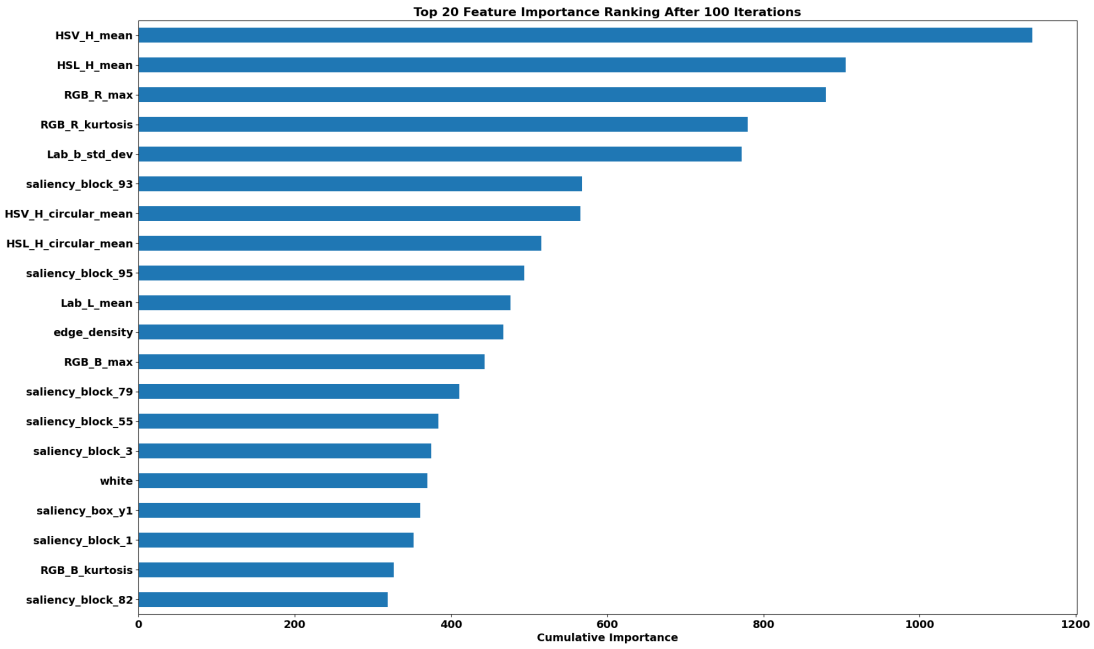


Figure 4.5: Top 20 most informative features for the Person category based on cumulative ranking after 100 CFR iterations.

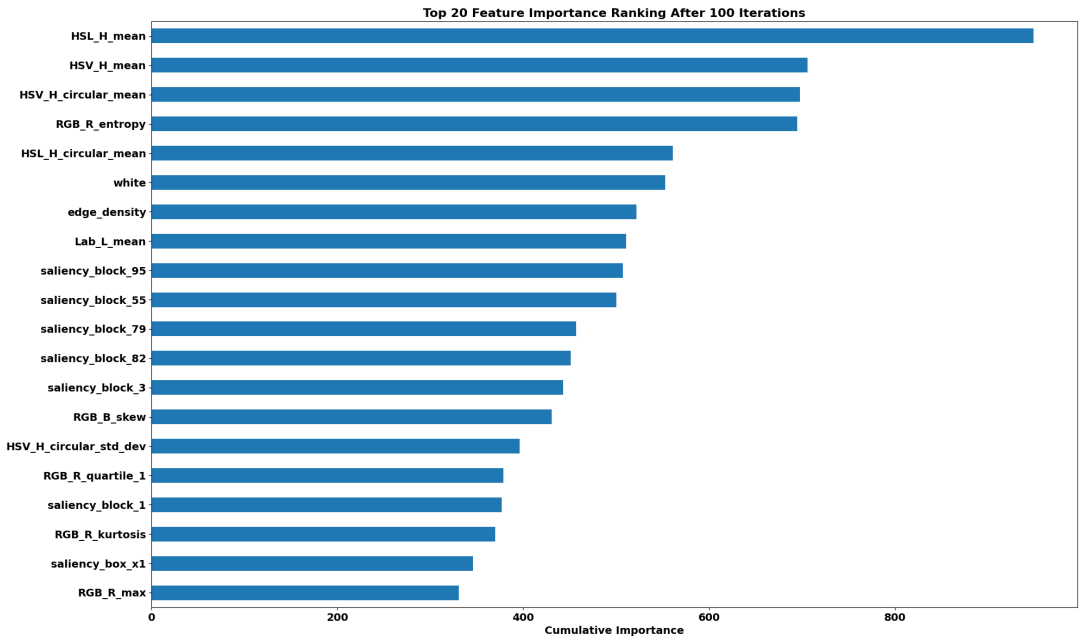


Figure 4.6: Top 20 most informative features for the Wheelchair category based on cumulative ranking after 100 CFR iterations.

tural saliency (e.g., *saliency_block_95*) were recurrently selected. These commonalities reinforce that balanced color distributions and stable structural context are strong, generalizable predictors for the ODD Checker to flag *Uncertain* cases accurately.

4.3.3.2 Statistical Analysis of Key Features.

Statistical details of the key features are summarized in Tables 4.2 and 4.3. For each feature, the range (Min–Max), central tendency (Mean), and dispersion (Std) are reported separately for *Certain* and *Uncertain* predictions. Systematic shifts in means and standard deviations can be interpreted as indicators of conditions that either stabilize detector outputs (higher certainty) or destabilize them (higher uncertainty). Since units differ across features, the analysis focuses on *directionality* and *relative separation* rather than absolute magnitudes.

Across both categories, hue-related features—*HSV_H_mean*, *HSL_H_mean*, and *HSV_H_circular_mean*—consistently exhibit higher means in *Certain* predictions, with lower relative dispersion compared to *Uncertain* predictions. This highlights the stabilizing role of balanced chromatic distributions in supporting reliable detections. In the Person category, additional separation is evident in red channel statistics (*RGB_R_max*,

RGB_R_kurtosis), suggesting that structured and well-distributed red channel values contribute to certainty in classification.

In contrast, *Uncertain* predictions are frequently associated with greater variability in intensity and structural descriptors. Features such as *edge_density*, *white*, and kurtosis measures display higher standard deviations in uncertain cases, reflecting the destabilizing influence of clutter, glare, or heavy-tailed intensity distributions. The Wheelchair category demonstrates stronger separation in entropy and luminance measures (*RGB_R_entropy*, *Lab_L_mean*), indicating that overly uniform chromatic distributions or excessively bright conditions are linked to reduced detection reliability.

Overall, color diversity and spatial saliency emerge as robust indicators of prediction certainty, while visual complexity—captured by skewness, kurtosis, or edge density—correlates with increased uncertainty. These statistical tendencies form the basis for the rule thresholds identified by the ODD Checker, providing a quantitative explanation for its ability to identify *Uncertain* cases. Detailed numeric distributions for both categories are presented in Tables 4.2 and 4.3.

Table 4.2: Statistical analysis of the top 20 most discriminative features for the Person category.

Feature	**Certain Prediction**				**Uncertain Prediction**			
	Min	**Max**	**Mean**	**Std**	**Min**	**Max**	**Mean**	**Std**
HSV_H_mean	0.671	132.990	58.588	25.671	0	123.030	19.808	21.184
HSL_H_mean	0.671	132.813	58.580	25.665	0	122.952	19.813	21.176
RGB_R_max	29	255	194.885	80.421	33	255	239.266	41.396
RGB_R_kurtosis	-1.797	36.587	-0.763	1.489	-1.745	102.760	1.948	8.279
Lab_b_std_dev	0.323	21.971	4.149	2.629	0	17.907	4.404	2.596
saliency_block_93	3926	10229242	1245113.034	709728.234	0	4010418	1193072.664	735518.496
HSV_H_circular_mean	0.0005	179.990	108.826	71.163	0	179.999	109.654	84.692
HSL_H_circular_mean	0.006	179.999	108.832	71.151	0	179.999	109.997	84.612
saliency_block_95	3770	12862583	552209.845	633601.422	0	9737733	597741.945	526186.824
Lab_L_mean	2.605	250.980	104.318	58.437	5.421	252.854	118.229	64.493
edge_density	0	0.059	0.017	0.011	0	0.062	0.027	0.014
RGB_B_max	29	255	192.589	80.876	29	255	238.445	42.069
saliency_block_79	0	4864017	511250.815	517662.737	0	3881974	460920.355	490101.094
saliency_block_55	5290	10167229	510621.735	610563.091	14801	10945639	634674.871	753748.602
saliency_block_3	3149	7646284	682049.062	532653.366	126	5344568	529355.908	433379.560
white	0	0.948	0.124	0.203	0	0.976	0.226	0.304
saliency_box_y1	0	543	72.239	82.427	0	596	79.917	83.034
saliency_block_1	0	4079486	373917.907	356635.229	0	3710982	325755.689	386713.093
RGB_B_kurtosis	-1.763	24.245	-0.727	1.312	-1.719	83.269	1.681	7.143
saliency_block_82	0	9692714	523653.133	641494.480	0	8508496	454948.867	453631.226

4.3.3.3 Decision Tree Rules and ODD Checker.

With the top-ranked features identified, DT models were employed to extract interpretable decision rules that define the operational boundaries of the ODD Checker. While statisti-

Table 4.3: Statistical analysis of the top 20 most discriminative features for the Wheelchair category.

Feature	Certain Prediction				Uncertain Prediction			
	Min	Max	Mean	Std	Min	Max	Mean	Std
HSL_H_mean	0.234	132.813	55.742	28.953	0	127.523	20.392	22.159
HSV_H_mean	0.234	132.990	55.750	28.962	0	127.521	20.384	22.162
HSV_H_circular_mean	0.001	179.999	108.739	71.171	0	179.999	108.576	84.755
RGB_R_entropy	0.677	5.423	4.440	0.958	0.544	5.418	3.497	1.271
HSL_H_circular_mean	0.002	179.999	108.745	71.157	0	179.999	109.051	84.649
white	0	0.950	0.134	0.209	0	0.976	0.211	0.294
edge_density	0	0.056	0.018	0.012	0	0.062	0.029	0.014
Lab_L_mean	2.605	249.900	106.803	55.705	4.655	252.850	114.535	63.886
saliency_block_95	3770	11373720	571377.527	669824.398	50137	12862583	621487.509	598927.934
saliency_block_55	5290	10167229	580297.015	697217.488	21687	7672595	660811.351	638618.256
saliency_block_79	0	4864017	498801.541	515531.571	0	3128889	500541.410	498634.634
saliency_block_82	0	9692714	496334.335	649625.686	0	8508496	466065.462	556524.204
saliency_block_3	126	5628633	670912.900	543108.368	17067	7646284	599574.447	504050.277
RGB_B_skew	-4.825	2.283	0.154	0.730	-8.485	2.261	-0.069	1.509
HSV_H_circular_std_dev	1.628	88.897	36.600	17.335	0	80.182	16.471	13.996
RGB_R_quartile_1	0	255	64.617	52.773	0	255	76.224	76.042
saliency_block_1	0	4368944	375995.212	423719.114	0	3654900	316305.985	377691.707
RGB_R_kurtosis	-1.797	30.203	-0.533	2.106	-1.645	89.461	1.890	8.277
saliency_box_x1	0	1079	129.996	182.963	0	921	140.586	164.033
RGB_R_max	30	255	210.013	73.226	29	255	236.121	48.781

cal comparisons provide valuable descriptive insights into feature distributions, they are insufficient for classification with high reliability. Decision Trees address this by combining multiple features into rule-based boundaries that separate *Certain* from *Uncertain* predictions. These rules form the backbone of the ODD Checker, enabling the system to dynamically assess input images based on feature conformity. The use of DTs aligns with XAI principles by providing transparency and interpretability in the decision-making process.

A representative subset of decision rules learned for the Person category is presented below:

IF (HSV_H_mean > 26.107) **AND** (Lab_L_mean ≤ 203.556) **AND** (saliency_block_54 ≤ 2060165.0) **AND** (saliency_block_12 > 110291.0) **THEN** **Certain Prediction**;

IF (HSV_H_mean ≤ 26.107) **AND** (RGB_B_max > 150.0) **AND** (saliency_block_93 > 208535.0) **AND** (saliency_block_46 ≤ 3280024.0) **THEN** **Uncertain Prediction**;

IF (HSV_H_mean > 26.107) **AND** (Lab_L_mean ≤ 203.556) **AND** (saliency_block_54 ≤ 2060165.0) **AND** (saliency_block_12 ≤ 110291.0) **THEN** **Certain Prediction**;

These rules highlight hue (*HSV_H_mean*) as the primary discriminative factor for persons, consistently separating stable from unstable cases. Luminance thresholds (*Lab_L_mean*) further stabilize predictions, while saliency-based conditions

(*saliency_block_54*, *saliency_block_12*) capture structural variations in cluttered environments. The combination of chromatic balance and luminance control thus supports confident detection of person instances.

A representative subset of decision rules learned for the Wheelchair category is presented below:

IF (HSV_H_mean > 24.511) **AND** (RGB_R_entropy > 3.183) **AND** (HSL_H_circular_mean > 11.586) **AND** (saliency_block_79 ≤ 460637.5) **AND** (RGB_R_entropy > 4.26) **THEN** **Certain Prediction**;

IF (HSV_H_mean ≤ 24.511) **AND** (RGB_B_max > 150.0) **AND** (saliency_box_size ≤ 0.813) **AND** (saliency_box_x1 ≤ 423.0) **AND** (saliency_block_33 ≤ 2053392.0) **THEN** **Uncertain Prediction**;

IF (HSV_H_mean > 24.511) **AND** (RGB_R_entropy > 3.183) **AND** (HSL_H_circular_mean > 11.586) **AND** (saliency_block_79 > 460637.5) **AND** (RGB_R_entropy ≤ 5.381) **THEN** **Certain Prediction**;

These rules show that hue remains a dominant separator, but entropy (*RGB_R_entropy*) and circular hue measures (*HSL_H_circular_mean*) provide decisive refinement for wheelchair detection. Saliency-based indicators (*saliency_block_79*, *saliency_block_33*) capture structural variations in complex indoor settings, while luminance-related conditions ensure robustness across brightness changes. This interplay of entropy and saliency yields a reliable framework for distinguishing uncertain wheelchair predictions.

Overall, the decision rules for both categories demonstrate how statistical patterns in chromatic, luminance, entropy, and saliency features are translated into operationalizable safety checks. Person detection relies more heavily on luminance stabilization, whereas wheelchair detection depends more on entropy and structural saliency. Embedding these interpretable boundaries into the ODD Checker provides a transparent, data-driven mechanism for real-time monitoring of prediction certainty across diverse indoor mobility scenarios.

4.3.3.4 Performance Evaluation of the ODD Checker.

The performance of the ODD Checker was assessed on the 40% of the dataset reserved for evaluation, using confusion matrices for both the Person and Wheelchair categories (Table 4.4). The evaluation captures the system's ability to distinguish between *Certain* and *Uncertain* predictions across all cases, with particular attention to how reliably it flags *Uncertain* predictions when the Object Detector produces misclassifications. The evaluation metrics are defined as follows:

True Positive: The Object Detector misclassifies objects in the image, and the ODD checker successfully identifies the *Uncertain* prediction.

False Positive: The Object Detector correctly classifies the objects in the image, but the ODD checker incorrectly identifies the prediction as *Uncertain*.

True Negative: The Object Detector correctly classifies the objects in the image, and the ODD checker identifies the prediction as *Certain*.

False Negative: The Object Detector misclassifies objects in the image, but the ODD checker fails to identify the prediction as *Uncertain*.

False Positive Rate (FPR): The proportion of images where the ODD checker incorrectly identifies uncertainty (FP) out of all correctly classified images:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{4.2}$$

False Negative Rate (FNR): The proportion of images where the ODD checker fails to identify uncertainty (False Negative (FN)) despite the Object Detector misclassifying objects:

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{4.3}$$

Clopper-Pearson Interval: To enhance the evaluation, we incorporated the Clopper-Pearson interval, which provides a confidence interval for the Misclassification Rate. The Clopper-Pearson interval is calculated as:

$$\text{Clopper-Pearson} = \text{MSE} + \sqrt{\frac{\ln(1/\delta)}{2n}} \tag{4.4}$$

where MSE is the mean squared error, n is the test size, and δ is set to 0.1. This metric offers insight into the consistency and reliability of the system's performance, particularly for edge cases or datasets with variability. A smaller interval suggests greater confidence in the reported metrics, indirectly supporting the model's robustness under similar conditions.

For the Person category, the results in Table 4.4 indicate strong performance, with 2995 TNs and 1980 TPs. The system achieved an FPR of 8.58% and an FNR of 13.46%, highlighting effective identification of *Certain* predictions with minimal false alarms. The

Table 4.4: Confusion matrix values and error rates of the ODD Checker for the Person and Wheelchair categories.

Category	True Negative (TN)	FP	FN	True Positive (TP)	FPR	FNR
Person	2995	281	308	1980	8.58%	13.46%
Wheelchair	2502	346	270	775	12.15%	25.84%

Clopper-Pearson interval of 0.12 reflects high confidence in the reported metrics, supporting the reliability of the system's performance in distinguishing *Certain* and *Uncertain* predictions.

In the Wheelchair category, the results show that the system successfully identifies 2502 TNs and 775 TPs. However, the FPR of 12.15% and FNR of 25.84% indicate a relatively higher rate of misclassification compared to the Person category. The Clopper-Pearson interval of 0.18 provides confidence in the reliability of these metrics but also highlights variability, suggesting greater challenges with wheelchair-related images due to greater variation in object appearance or environmental conditions. These results reflect the inherent complexity of wheelchair-related scenes and underscore the importance of robust visual features in maintaining prediction certainty across diverse conditions.

4.3.3.5 Real-Time ODD Checker Performance Evaluation.

The real-time performance of the proposed ODD Checker was assessed using two 10-fps video sequences of indoor environments. The system's predictions were compared against manually annotated ground truth *Uncertain* labels. Table 4.5 summarizes the resulting confusion matrix values (TP, TN, FP, FN) for both Person and Wheelchair categories, while Figure 4.7 illustrates representative frames from the two scenarios, highlighting the difference between nominal and perturbed visual conditions.

The first video, recorded at CNR Villa in Genoa, contained only a few *Uncertain* instances, reflecting nominal indoor conditions in which the checker is well suited for practical deployment. The second video was generated by synthetically perturbing the same recording (e.g., brightness shifts) to artificially increase uncertainty and evaluate robustness under degraded conditions. Together, the two sequences provide complementary perspectives: the first shows baseline reliability in standard environments, while the second highlights system limitations under stress. Frame sequences for both videos are available in a public GitHub repository.*

*https://github.com/AhmadSamandari/XAIChecker_for_VideoAnalytics

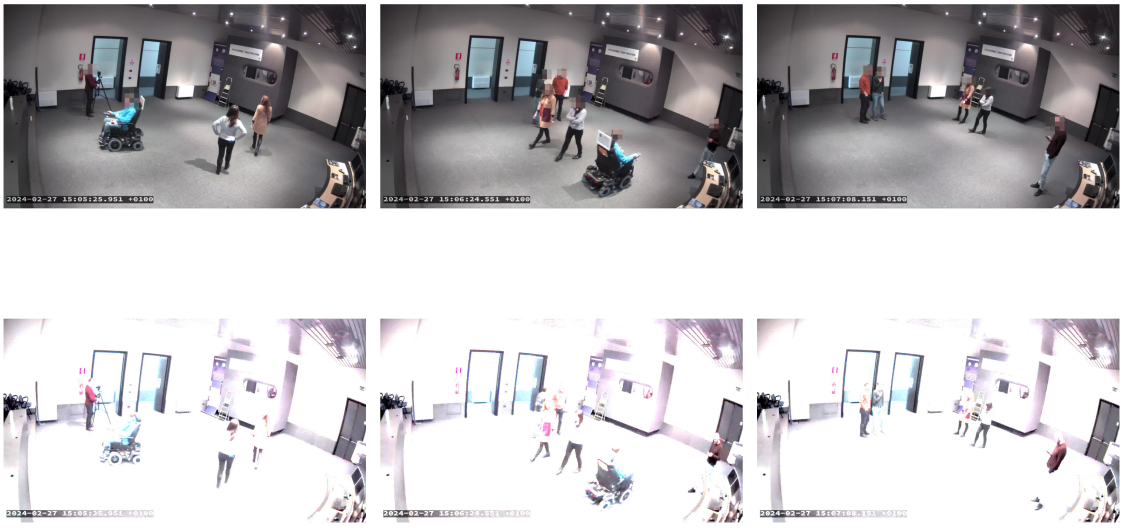


Figure 4.7: Representative frames from the two video sequences. Top row: Video 1 under nominal indoor conditions. Bottom row: Video 2 under synthetic perturbations (e.g., extreme brightness shifts).

Figures 4.8 and 4.9 present a frame-by-frame comparison of ODD Checker predictions against ground truth labels. In these plots, the green line represents ground truth annotations: a value of 1 indicates that the Object Detector failed to identify all present objects, corresponding to an *Uncertain* frame, while 0 indicates a *Certain* frame. The red line represents the ODD Checker output, where 1 denotes an *Uncertain* decision and 0 a *Certain* decision. Ground truth labels were derived manually by comparing the number of detected objects with the actual number of objects present in each frame.

In Video 1, mismatches between predictions and ground truth are sparse and isolated, suggesting that simple post-processing techniques such as temporal smoothing could further reduce occasional errors. In contrast, Video 2 shows frequent mismatches caused by visual degradation, reflecting the increased difficulty of maintaining reliable certainty assessment under perturbed conditions. These observations are consistent with the confusion matrix results in Table 4.5, where higher error counts are observed in the perturbed sequence, particularly for wheelchair detection. Despite this, the checker preserved responsiveness and continued to provide interpretable indications of uncertainty, demonstrating applicability both in nominal scenarios and in stress testing.

Table 4.5: Confusion matrix values for the ODD Checker in real-time evaluation across two video sequences.

Video	Category	Ground Truth Uncertain	TP	TN	FP	FN
Video 1	Person	4	0	554	0	4
	Wheelchair	6	0	552	0	6
Video 2	Person	487	485	0	71	2
	Wheelchair	219	152	144	195	67

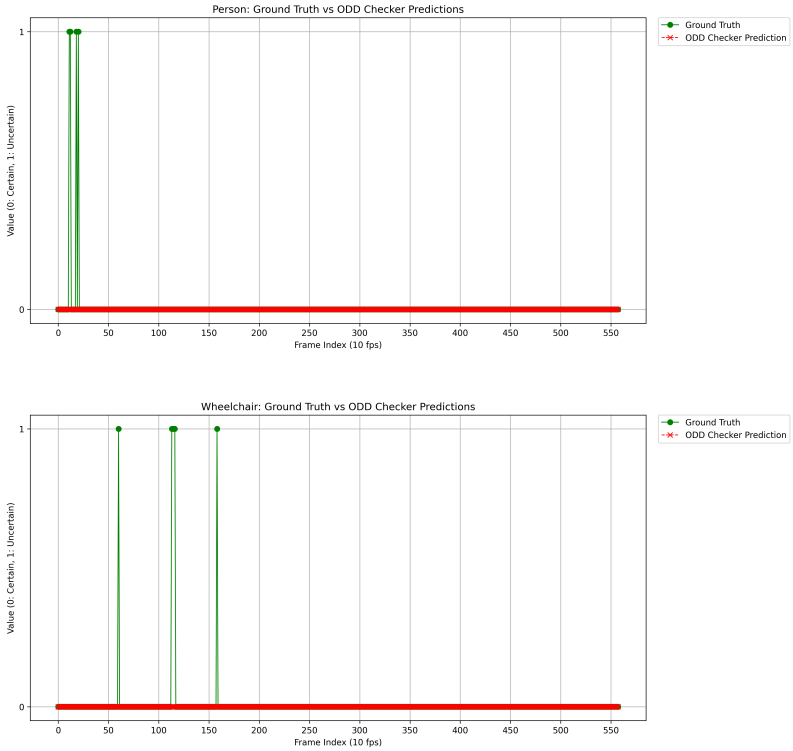


Figure 4.8: Video 1: Checker predictions vs. ground truth under nominal conditions.

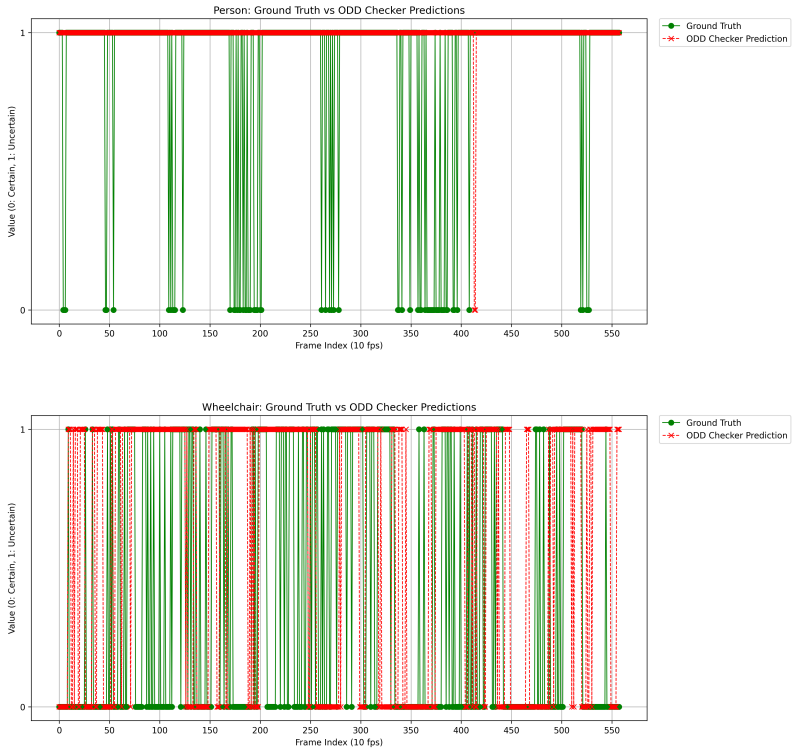


Figure 4.9: Video 2: Checker predictions vs. ground truth under perturbed conditions.

4.4 Conclusions

This chapter introduced an interpretable framework for Operational Design Domain (ODD) monitoring, designed to assess prediction certainty in object detection systems. The motivation stemmed from the need for reliable, transparent safety monitoring in assistive mobility applications, where both robustness and interpretability are critical. The proposed ODD Checker combines visual feature analysis with rule-based classification to provide a transparent layer of monitoring that operates independently of the internal variables of the object detector.

The framework was implemented using a dual-path approach: visual features were extracted from the detector's inputs and ranked through a cumulative feature ranking strategy, while decision tree models were trained to derive human-readable rules separating *Certain* from *Uncertain* predictions. This combination allows the Checker to capture relevant feature interactions while maintaining explicit, auditable decision boundaries. Statistical analysis highlighted the central role of hue- and luminance-related features, as well as saliency and texture complexity, in shaping prediction certainty across both Person and Wheelchair categories.

Evaluation on the 40% reserved dataset confirmed that the ODD Checker can reliably identify uncertainty. For the Person category, the Checker achieved a False Positive Rate (FPR) of 8.58% and a False Negative Rate (FNR) of 13.46%, indicating effective detection of uncertainty with minimal false alarms. For the Wheelchair category, performance exhibited greater variability (FPR 12.15%, FNR 25.84%), reflecting the inherent challenges of wheelchair detection, where visual diversity and environmental context lead to higher uncertainty. Nevertheless, the framework provided consistent and interpretable monitoring across both categories.

Real-time experiments further demonstrated applicability. Two 10-fps video sequences recorded at CNR Villa in Genoa were analyzed: one under nominal indoor conditions and another synthetically perturbed to simulate degraded visual environments. In nominal conditions, the Checker exhibited high stability, with sparse and isolated mismatches that could be mitigated by temporal smoothing. In the perturbed sequence, error rates increased as expected, yet the Checker preserved responsiveness and continued to provide explainable uncertainty alerts. Together, these evaluations confirmed both baseline reliability and limitations under stress, offering a realistic perspective on deployment readiness.

Beyond quantitative performance, a key contribution lies in transparency. The use of decision-tree rules enabled direct mapping between visual feature thresholds and prediction outcomes, transforming statistical patterns into operationalizable safety checks. This supports human understanding, facilitates debugging and adaptation to new environments, and aligns with emerging expectations for trustworthy AI.

Several limitations remain. The current framework operates on individual frames, without incorporating temporal dynamics that could smooth transient inconsistencies or leverage motion cues. In addition, reliance on handcrafted visual features may limit generalization across unseen environments with different lighting or structural conditions. Finally, while the Checker identifies uncertainty, it does not yet provide corrective mechanisms to recover from it.

Future work will therefore focus on two main directions. First, temporal extensions such as sliding-window aggregation or majority voting can be incorporated to stabilize predictions across sequences, reducing the impact of frame-level fluctuations. Second, adaptive correction modules can be developed so that, upon detecting an *Uncertain* prediction, the system applies targeted refinements aimed at shifting the input back into the *Certain* range. Such refinements may include low-level adjustments (e.g., brightness correction or noise filtering) as well as higher-level techniques such as counterfactual gen-

eration or autoencoder-based reconstruction, which can produce feature representations more consistent with the detector's operational domain.

In conclusion, this chapter demonstrated that a feature-driven, rule-based ODD Checker can provide reliable and transparent monitoring of prediction certainty in safety-critical mobility scenarios. By combining visual feature analysis, decision-tree rules, and real-time evaluation, the framework establishes a foundation for trustworthy ODD monitoring. The results confirm effectiveness under nominal conditions and resilience under perturbations, while outlining clear pathways for enhancement in subsequent chapters.

CHAPTER 5

Vision-Language Models for Scene Comprehension Support in Autonomous Wheelchair Navigation

5.1 Introduction

This chapter investigates explainable monitoring mechanisms based on Vision-Language Models (VLMs) for scene comprehension in autonomous mobility, with a particular emphasis on indoor smart mobility scenarios. The focus is on how VLMs can provide semantic safety assessments and human-readable explanations, complementing traditional perception pipelines in assistive navigation.

5.1.1 Background and Motivation

VLMs have recently emerged as a transformative paradigm in computer vision and robotics. Traditional pipelines for visual recognition—such as image classification, object detection, and semantic segmentation—rely heavily on supervised training with large annotated datasets and task-specific fine-tuning. While effective in structured settings, these approaches often struggle in dynamic or unstructured environments due to limited generalization and scalability.

VLMs overcome these limitations by learning joint vision–language representations from web-scale image–text pairs, enabling *zero-shot transfer* to downstream tasks without requiring additional labeled data [71, 82]. This paradigm shift has demonstrated remarkable performance across a wide spectrum of vision tasks by aligning visual features with natural language prompts.

In robotics, this capability is particularly compelling. Autonomous systems such as self-driving cars, drones, and assistive wheelchairs must operate safely in dynamic human environments. While conventional perception systems are well suited for detecting predefined hazards, they often fall short in reasoning about complex, ambiguous, or

unforeseen conditions. VLMs offer an avenue for semantic scene comprehension that aligns more closely with human judgment, providing a new layer of safety assurance.

5.1.2 Vision-Language Model (VLM): Principles and Architecture

VLMs integrate visual and textual information into a shared representation space, enabling joint reasoning across modalities. At a high level, a VLM consists of three core components: a vision encoder, a text encoder, and a fusion mechanism that aligns the two streams of information (Figure 5.1).

The vision encoder (e.g., convolutional networks or vision transformers) converts raw image inputs into dense feature embeddings that capture semantic content such as objects, textures, or spatial layout. The text encoder (e.g., transformer-based language models) maps natural-language prompts into a comparable embedding space, representing linguistic concepts and context. To align these two modalities, a projector or fusion layer is introduced, ensuring that visual and textual features interact meaningfully in a shared representation.

Several alignment strategies have been proposed in the literature. Contrastive approaches learn to maximize similarity between paired image–text inputs while minimizing similarity with mismatched pairs, resulting in a shared embedding space optimized for retrieval and classification. Generative approaches, in contrast, couple a visual encoder with a large language model, enabling open-ended reasoning by conditioning textual generation on visual context. Hybrid approaches extend these mechanisms with cross-attention layers that allow tokens from one modality to attend to representations from the other, improving fine-grained grounding.

Once aligned, these multimodal embeddings are processed by a large language model that performs reasoning and generates outputs grounded in both visual evidence and textual context [1]. Such an architecture enables a wide range of tasks, from image captioning and visual question answering to semantic safety assessment in autonomous systems.

5.1.3 Rare and Unforeseen Risks: The Swan Metaphor

A central challenge in safety assurance is that not all hazardous conditions can be exhaustively enumerated. This has been captured by the metaphor of *swans*:

> **Grey swans** represent rare but partially understood risk conditions that can be revealed through systematic testing.

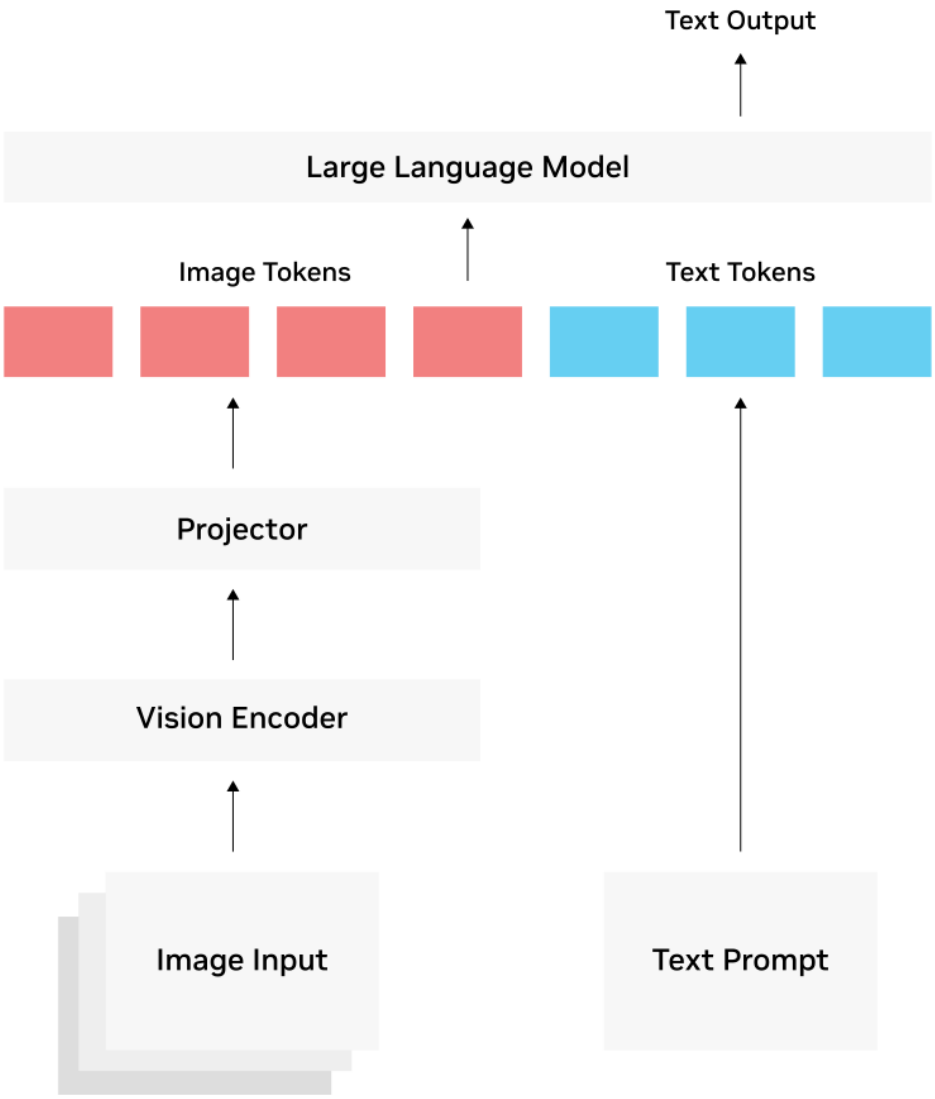


Figure 5.1: Conceptual architecture of a VLM. Image inputs and text prompts are encoded into a shared representation space, where they are jointly processed for multimodal reasoning [1].

> **Black swans** represent fundamentally unknowable risks that cannot be anticipated or exhaustively specified [87].

Figure 5.2 illustrates this taxonomy. Grey swans capture conditions that, while not frequent, can be analyzed and incorporated into testing pipelines. Black swans, by contrast, represent hazards that remain outside the scope of structured evaluation frameworks, challenging the completeness of safety assurance.

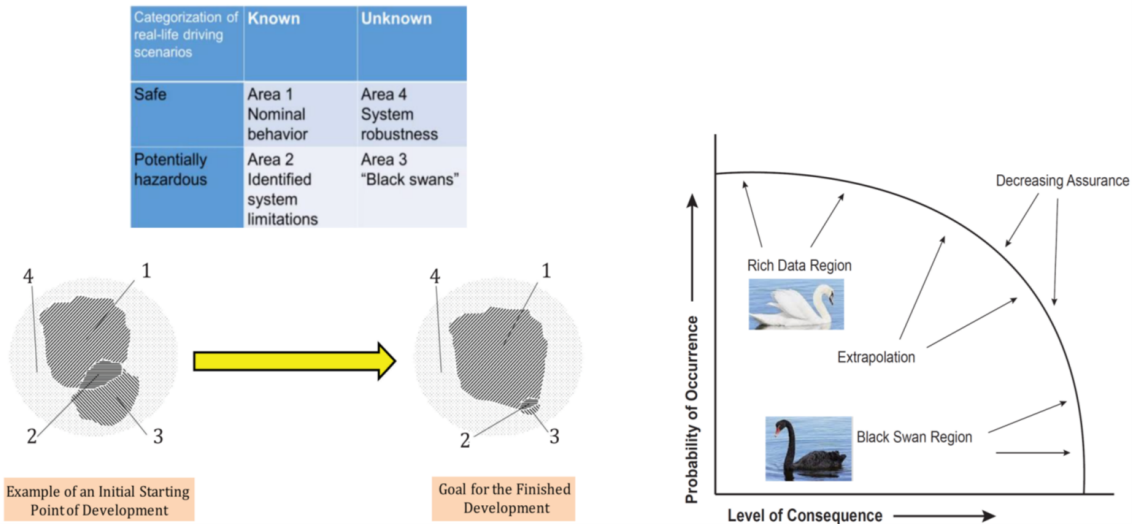


Figure 5.2: Swans in risk analysis. Grey swans refer to rare but partially understood risk conditions. Black swans represent fundamentally unknowable hazards that may remain invisible to structured evaluation frameworks.

VLMs are especially well positioned to uncover grey swan cases. Through their broad visual–textual grounding and zero-shot reasoning capabilities, they can highlight borderline scenarios that may otherwise be overlooked by rule-based detectors. For example, VLMs can identify ambiguous conditions such as dimly lit stairways or semi-crowded corridors that are difficult to classify as strictly safe or unsafe. In this way, they extend the safety evaluation frontier by directing human attention to edge cases where system vulnerabilities are most likely to surface.

5.1.4 Contributions of This Chapter

This chapter integrates the above perspectives and provides a systematic evaluation of VLMs for scene-level safety assessment in autonomous wheelchair navigation. The specific contributions are:

1. **Explainable safety filtering** – VLMs are introduced as semantic and interpretable classifiers that complement traditional perception systems in assistive robotics.

2. **Integration of the swan metaphor** – VLM outputs are contextualized as tools for surfacing rare and ambiguous risks, providing a conceptual bridge between abstract safety assurance and practical deployment.

3. **Benchmarking study** – A systematic comparison is performed across three representative models—Contrastive Language-Image Pre-Training (CLIP) [71], MiniGPT-4 [74], and GPT-4 with Vision (GPT-4V) [75]—on a curated dataset of 89 indoor scenes, extended with synthetic perturbations (fog, smoke, and reduced illumination). The classification task distinguishes between *Safe to Proceed* and *Risky to Proceed* based on four safety factors: immediate obstacles, crowd density, visibility degradation, and stair-related fall risks. The evaluation highlights the trade-offs between reasoning power, inference efficiency, and robustness under degraded conditions.

5.2 Methodology

5.2.1 Model Architectures and Reasoning Pipelines

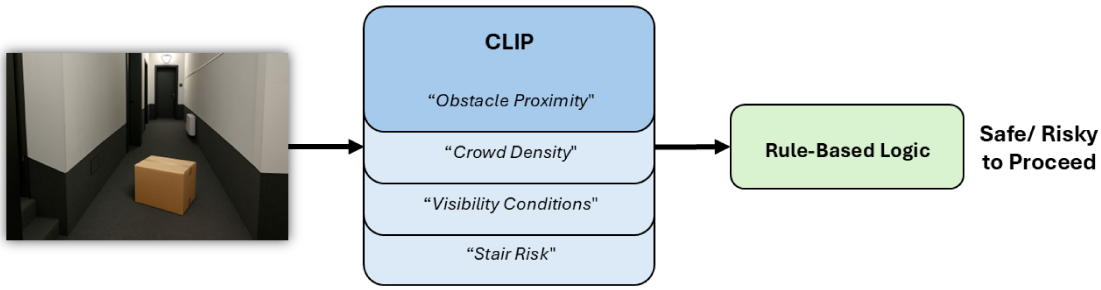
This study evaluates three VLMs selected to represent complementary trade-offs between reasoning depth, inference latency, and deployability: CLIP (ViT-B/32), MiniGPT-4, and GPT-4V. All three were configured to perform binary scene-level classification under a unified safety framework. The task was to classify an indoor scene as either *Safe to Proceed* or *Risky to Proceed* based on four predefined safety dimensions: obstacle proximity, crowd density, visibility degradation, and stair-related fall risk.

Figure 5.3 provides an overview of the reasoning pipelines implemented for each model.

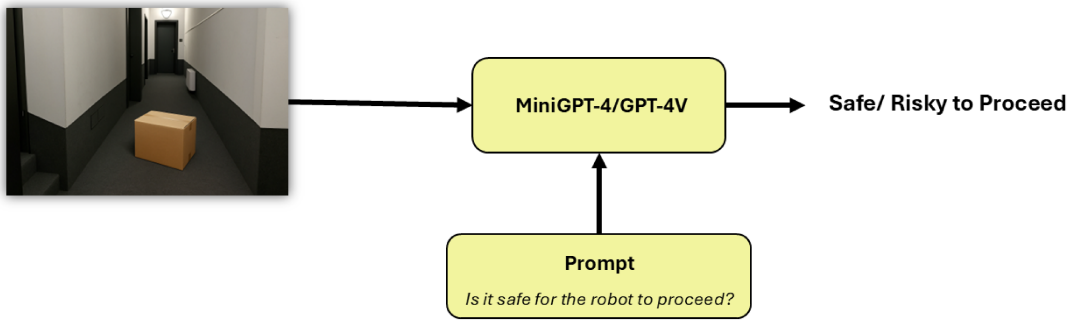
5.2.1.1 CLIP (Contrastive Language–Image Pretraining)

CLIP [71] is a VLM developed by OpenAI that learns joint representations of images and text through contrastive training. During pretraining, CLIP was exposed to hundreds of millions of image–text pairs collected from the web. Its objective is to align an image and its paired caption closely in the embedding space while pushing apart mismatched pairs. This alignment enables CLIP to generalize in a zero-shot fashion to downstream tasks without supervised retraining.

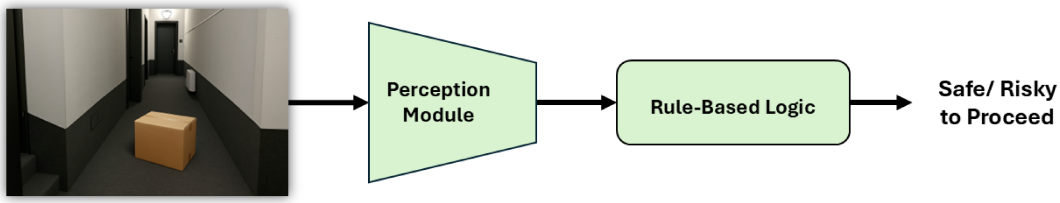
For this study, the ViT-B/32 variant of CLIP was used. A modular, multi-branch design was implemented, where four parallel modules were created, each dedicated to one of the safety dimensions (obstacles, crowd, visibility, stairs). Each module computed cosine similarity between the encoded image and a dedicated label set of textual phrases. The



(a)



(b)



(c)

Figure 5.3: Model architectures and reasoning pipelines. (a) Modular CLIP with rule-based fusion of safety cues. (b) Decision-making using MiniGPT-4 and GPT-4V. (c) Traditional two-stage perception pipeline.

outputs were then fused with simple rule-based logic: if any module detected a high-risk condition, the scene was classified as *Risky to Proceed*; otherwise, it was classified as *Safe to Proceed*.

This modular design improved interpretability by isolating the contribution of each safety factor and reduced errors caused by overlapping cues (for instance, high crowd density combined with low visibility). CLIP's efficiency and low computational footprint make it suitable for embedded or CPU-only systems, although its reasoning ability is limited by the expressiveness of its label sets.

5.2.1.2 MiniGPT-4

MiniGPT-4 [74] is an open-source VLM that combines a pretrained vision encoder (BLIP-2) with a medium-scale language model (Vicuna-7B). Unlike contrastive models such as CLIP, MiniGPT-4 follows a generative approach: visual features are projected into the token space of the language model, which then generates text conditioned on both the image and the input instruction.

This design enables MiniGPT-4 to perform higher-level reasoning compared to CLIP and to provide human-readable textual rationales for its decisions. It operates fully offline on local machines and is accessible for research use. However, it is computationally heavier and slower, particularly on CPU-only systems, and its generative nature can lead to hallucinations, where hazards not present in the scene are reported.

5.2.1.3 GPT-4 with Vision (GPT-4V)

GPT-4V [75] is the multimodal extension of the Generative Pre-trained Transformer 4 (GPT-4) large language model. It incorporates advanced attention mechanisms that allow visual features to interact with language tokens across multiple layers, enabling high-level multimodal reasoning. This enables GPT-4V to capture spatial relations and contextual dependencies more effectively than either CLIP or MiniGPT-4.

In this study, GPT-4V was accessed through the OpenAI API (November 2023 version). Compared to MiniGPT-4, GPT-4V provided more consistent semantic reasoning and produced fewer hallucinations, but at the cost of higher inference latency and reliance on cloud-based execution. The inclusion of GPT-4V in the study served to establish an upper bound on performance for scene-level safety classification.

In summary, these three VLMs were selected to span a spectrum from lightweight, efficient embedding-based reasoning (CLIP), to medium-scale generative reasoning (MiniGPT-4),

and advanced multimodal semantic understanding (GPT-4V). Their comparison illustrates the trade-offs between interpretability, computational efficiency, and reasoning capability in the context of scene-level safety monitoring.

5.2.2 Prompt Engineering and Configuration Tuning

Prompt engineering plays a central role in adapting VLMs to specific downstream tasks. Since VLMs are pretrained on diverse web-scale data rather than curated datasets, their outputs are highly sensitive to the formulation of input prompts. The wording of labels, the structure of instructions, and the specification of output format can strongly influence both classification accuracy and the consistency of generated explanations. Prompt engineering can therefore be understood as the design of task-specific textual inputs—either discrete templates or instructions—that guide a model's multimodal reasoning without modifying its parameters [167].

In this study, two prompting strategies were adopted, corresponding to the underlying model architectures: (i) label-based prompting for CLIP, and (ii) instruction-based prompting for MiniGPT-4 and GPT-4V. The first approach corresponds to the category of *hard prompts*, in which fixed textual phrases are compared to image embeddings. The second corresponds to *instruction prompting*, in which natural-language directives are provided to constrain the model's generative reasoning.

5.2.2.1 CLIP

CLIP was configured using a modular approach in which each safety factor was associated with a dedicated label set of textual phrases. These phrases were carefully designed to minimize semantic overlap and to represent distinct conditions within each safety dimension:

Obstacle proximity: "No obstacles in front", "Obstacles far away", "Immediate obstacles in front (very close)".

Crowd density: "No pedestrians nearby", "Low crowd", "High crowd".

Visibility conditions: "Crisp and clear image", "Scene covered in fog or smoke, objects faded or obscured".

Stair risk: "Flat corridor, no stairs", "Visible stairs close ahead", "Patterned or dark floor but flat".

Each branch produced a similarity score between the encoded image and its associated labels. The results were then fused through rule-based logic: if any module matched a high-risk condition, the final output was "*Risky to Proceed*"; otherwise, "*Safe to Proceed*".

5.2.2.2 MiniGPT-4 and GPT-4V

MiniGPT-4 and GPT-4V required structured natural-language instructions rather than discrete label sets. To enforce consistency and reduce ambiguity, a comprehensive prompt was designed with explicit task specifications, safety rules, and output formatting constraints. The full instruction is reported below:

```
You are installed on a mobile robot and it is the vision
of robot, you are navigating this robot forward.

Task:
- Output only one label: "Safe to proceed" or "Risky to
proceed".
- Base your decision ONLY on the risk of immediate collision,
falling, or poor visibility.

Rules:
- If the forward path is clear and visibility is good -
say "Safe to proceed".
- If any person, wheelchair, robot, table, or other
obstacle is immediately in the forward path - say "Risky
to proceed".
- Immediate means close enough that you cannot safely
stop or change direction without collision.
- If there is heavy fog, heavy smoke, or anything that
seriously reduces visibility - say "Risky to proceed",
even if no obstacle is visible.
- Ignore people, objects, or robots that are far away,
not in your immediate path, or not posing an urgent
risk.
- Treat small groups of people standing close in your
forward path as obstacles.
- Do NOT assume people will move aside.
```

```
Also briefly explain why you made your decision (example:
"person directly blocking the way", "heavy smoke blocking
vision").

Answer format:
Decision: [Safe to proceed / Risky to proceed]
Reason: [short explanation]
```

The design of this prompt served several objectives. The *Task* and *Rules* sections constrained the model to focus on immediate hazards, thereby reducing false alarms from distant or irrelevant objects. The inclusion of visibility-related conditions ensured that abstract risks such as fog and smoke were accounted for alongside physical obstacles. The requirement for a concise rationale encouraged the generation of human-readable explanations, strengthening traceability and explainability. Finally, the fixed *Answer format* standardized outputs across models, facilitating automatic parsing and evaluation.

It should be noted that while prompt engineering improves task alignment, recent studies have highlighted that it also introduces new challenges, including robustness to distribution shifts and susceptibility to adversarial manipulations [167]. These considerations are particularly important when deploying VLMs in safety-critical settings.

5.2.3 Implementation Details

CLIP (ViT-B/32) and MiniGPT-4 (Vicuna-7B) were executed locally on a CPU-only Linux workstation. CLIP was run using `open_clip` and Hugging Face implementations; MiniGPT-4 was deployed with BLIP-2 integration. GPT-4V was accessed through the OpenAI API (November 2023 version) [75].

All models were evaluated on the same dataset with unified task definitions. CLIP outputs were obtained via cosine similarity, while MiniGPT-4 and GPT-4V produced classification and rationales via prompts. This ensured comparability across architectures in terms of task framing and evaluation.

5.3 Experimental validation

5.3.1 Dataset

The evaluation employed a dataset of 89 unique indoor scenes designed to capture safety-relevant conditions in assistive wheelchair navigation. A total of 73 high-resolution images (1920×1080) were collected at the CNR Villa in Genoa, covering corridors and communal areas with varied obstacles (chairs, tables, wheelchairs), different crowd densities, and occasional staircases. To complement these, 10 additional corridor scenes (309×240) without obstacles were sourced from the KTH-IDOL2 dataset [168].

To assess robustness under degraded conditions, 6 scenes were synthetically perturbed using DALLE-based editing tools*. Perturbations included fog overlays, smoke effects, and reduced brightness, preserving scene semantics while introducing realistic visibility challenges.

All images were manually annotated along four safety dimensions: immediate obstacles, crowd density, visibility degradation, and fall risk from stairs. Based on annotations, 35 scenes were labeled as *Safe to Proceed* and 54 as *Risky to Proceed*. Representative examples are shown in Figure 5.4.

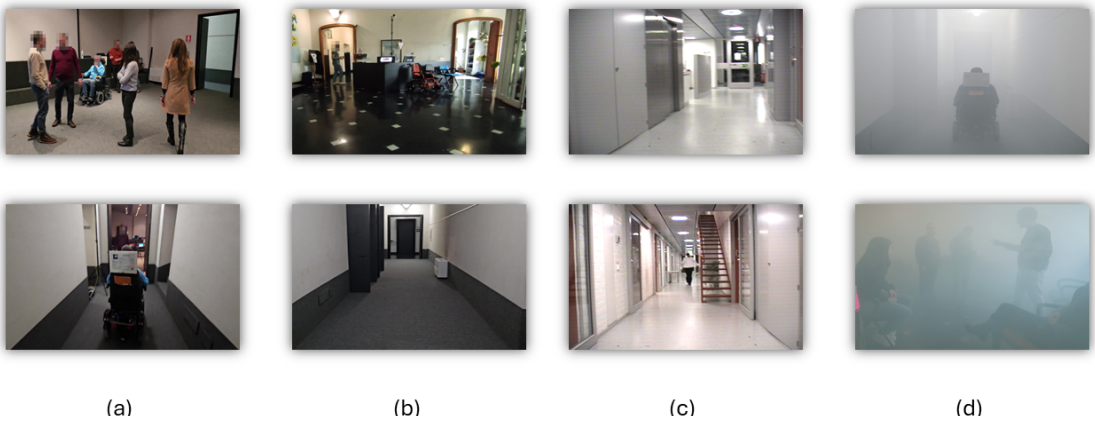


Figure 5.4: Representative dataset examples: (a) risky scenes with obstacles or high crowd density; (b) safe scenes with clear forward paths; (c) safe scenes from KTH-IDOL2; (d) perturbed scenes with fog or smoke, labeled as risky due to impaired visibility.

*https://openai.com/dall-e

5.3.2 Results

The evaluation considers CLIP (ViT-B/32), GPT-4V, and MiniGPT-4 on binary scene-level safety classification. CLIP was implemented as a four-branch modular architecture that independently assessed obstacle proximity, crowd density, visibility degradation, and stair-related fall risk, with outputs fused by rule-based logic. Model outputs (decision and accompanying explanation) were reviewed to assess alignment with ground truth and reasoning consistency; discrepancies between predicted labels and justifications were flagged as reasoning errors. A summary of comparative performance is reported in Table 5.1.

Table 5.1: Performance of vision–language models on scene-level safety classification. TN: True Negatives, FP: False Positives, FN: False Negatives, TP: True Positives.

Model	TN	FP	FN	TP	Accuracy	Precision	Recall	F1-Score
CLIP	21	14	10	44	73.0%	75.9%	81.5%	78.6%
GPT-4V	33	2	2	52	95.5%	96.3%	96.3%	96.3%
MiniGPT-4	12	22	8	46	65.2%	67.6%	85.2%	75.1%

CLIP. CLIP achieved 73.0% accuracy, 75.9% precision, 81.5% recall, and a 78.6% F1-score (Table 5.1). Most false negatives occurred in scenes with ambiguous spatial cues (e.g., nearby wheelchairs interpreted as distant). False positives were often due to overestimating risk in empty corridors or misjudging depth and proximity. Occasional errors included patterned floors misidentified as stairs and image-quality issues conflated with visibility degradation. Figure 5.5 shows a case labeled risky by CLIP with a misattributed cause. Despite these limitations, the modular design offered fast, interpretable, and resource-efficient inference (average ∼0.33 s per image on CPU-only hardware), suitable for embedded or real-time use.

GPT-4V. GPT-4V attained the highest performance across all metrics (95.5% accuracy; 96.3% recall), consistently identifying obstacle proximity, visibility degradation, and crowd-related risks with strong semantic reasoning. Occasional misclassifications occurred (Figure 5.5), possibly reflecting user-centric rather than system-centric navigation assumptions. High computational cost and API-only availability limit applicability in latency- or privacy-constrained settings; inference time ranged from 2 to 58 s per image depending on resolution and API latency.



Figure 5.5: Scene labeled as *Risky to Proceed* by MiniGPT-4 and CLIP; CLIP's rationale was incorrect. GPT-4V misclassified the scene as *Safe to Proceed*.

MiniGPT-4. MiniGPT-4 achieved 85.2% recall but lower precision (67.6%) and overall accuracy (65.2%). False positives included spurious pedestrian detections, while false negatives involved missed obstacles. In Figure 5.5, it identified risk but with a partially correct rationale. The model was sensitive to visible hazards yet prone to over-detection; CPU-only inference time ranged from 12 to 84 s per image, the slowest among the three.

Figure 5.6 illustrates a borderline case misclassified by all models and corresponding to a *grey swan* as defined in Section 5.1 (rare but partially understood risk conditions). The scene contains a pedestrian and a staircase at short distance but was annotated as *Safe to Proceed*. CLIP likely erred due to image resolution and ambiguous spatial cues; GPT-4V cited a person too close to avoid, likely overestimating distance; MiniGPT-4 incorrectly inferred heavy fog and a person despite clear visibility. Such cases motivate targeted data augmentation (e.g., additional stair configurations) to improve navigation awareness.

Figure 5.7 shows a *black swan* instance (unforeseeable or fundamentally underspecified conditions, per the introduction). A person in a wheelchair near the forward path creates ambiguity: although no direct obstacle exists, the path is not fully open. Annotators debated the correct label, and all three models flagged risk. A conservative decision is arguably appropriate; however, excessive false positives would lead to unnecessary braking and a degraded driving experience. Expanding coverage of such ambiguous layouts in the dataset is likely to reduce both over- and under-reactions.



Figure 5.6: Borderline scene with no immediate obstacles or crowd, misclassified by all models.



Figure 5.7: Ambiguous scenario: a person in a wheelchair near the forward path introduces uncertainty in an otherwise semi-open environment.

Summary. CLIP delivered competitive accuracy and interpretability with fast inference and full offline deployability, suitable for embedded use. GPT-4V achieved the strongest semantic robustness but requires cloud access and exhibits higher latency. MiniGPT-4 offered high sensitivity to hazards but suffered from over-prediction and slow processing. Overall, results expose trade-offs between semantic reasoning strength, speed, precision, and the feasibility of real-time deployment.

5.4 Conclusion

This chapter presented a benchmarking study of VLMs for explainable scene-level safety assessment in autonomous wheelchair navigation. By comparing three representative models—CLIP, MiniGPT-4, and GPT-4V—on a curated dataset of real and synthetically perturbed indoor scenes, the analysis assessed their capacity to classify environments as either *Safe to Proceed* or *Risky to Proceed* based on four safety dimensions: obstacle proximity, crowd density, visibility degradation, and stair-related fall risks.

The comparative evaluation highlighted clear trade-offs between interpretability, accuracy, and deployability. CLIP, with its contrastive architecture and modular label-based design, offered high interpretability and the fastest inference times on CPU-only systems. These characteristics make it suitable for embedded applications where transparency and efficiency are critical. However, its reliance on handcrafted label sets constrained its reasoning capacity, particularly in ambiguous or visually degraded conditions. GPT-4V achieved the highest overall performance, delivering superior accuracy and robustness in semantic reasoning. Its ability to consistently align with human risk perception demonstrates the potential of advanced multimodal transformers for safety-critical tasks. Nonetheless, the reliance on API-based inference, coupled with latency ranging from seconds to tens of seconds per image, restricts its applicability in real-time or privacy-sensitive contexts. MiniGPT-4 occupied an intermediate position. Its generative architecture, combining BLIP-2 with Vicuna-7B, enabled strong sensitivity to hazards and the ability to produce human-readable rationales. Yet, the model exhibited a high false positive rate and suffered from slow inference on CPU-only systems, limiting its viability for time-critical deployment.

Beyond raw performance metrics, this study underscored the importance of explainability in VLM-based safety assessment. MiniGPT-4 and GPT-4V, by producing short natural-language rationales, provided transparency into the reasoning process and en-

abled post-hoc auditing of decisions. Such capabilities are essential for trust in assistive robotics, where users, caregivers, and regulators require clear justifications for safety-related decisions. The integration of the swan metaphor offered a conceptual framework for interpreting rare and ambiguous risk cases. Grey swan conditions, such as dimly lit stairways or semi-crowded corridors, revealed the limits of rule-based detection while demonstrating the utility of VLMs for surfacing edge cases. Black swan scenarios, by contrast, emphasized the existence of fundamentally under-specified risks that challenge the completeness of structured evaluation frameworks. The presence of such cases in the dataset highlights the value of VLMs as semantic filters that can direct human attention to uncertain or borderline environments.

Several limitations of the present study should be acknowledged. The dataset comprised static images rather than video sequences, limiting the ability to evaluate temporal reasoning and motion-related risks. Inference was performed under CPU-only conditions, which, while representative of resource-constrained deployments, accentuates latency issues and may understate performance achievable with GPU acceleration. The evaluation also focused on four predefined safety dimensions; additional factors such as lighting dynamics, reflective surfaces, or occlusion patterns could further influence navigation safety but were not explicitly modeled here.

Building on these findings, several avenues for future research emerge. Hybrid pipelines that combine CLIP's efficiency with GPT-4V's semantic depth, possibly mediated by lightweight rule-based post-processing, offer a promising path toward balancing accuracy, interpretability, and deployability. Expanding datasets to include temporally sequenced video data would enable the assessment of temporal consistency and the anticipation of dynamic hazards. Incorporating simulation environments could provide controlled exposure to rare or hazardous conditions that are difficult to capture in real-world data collection. Another important direction involves refining prompt engineering strategies. Adaptive or learnable prompts may improve robustness across diverse environments and reduce susceptibility to distribution shifts or adversarial manipulation. Finally, integrating uncertainty quantification methods, such as conformal prediction or Bayesian reasoning, with VLM outputs could enhance safety by providing calibrated confidence estimates alongside binary decisions.

In conclusion, this chapter demonstrated that VLMs represent a promising step toward explainable and trustworthy AI in assistive navigation. While current models do not yet meet the latency and reliability demands of real-time autonomous mobility, they provide strong value for offline safety auditing, post-hoc validation, and the identification of rare

or ambiguous risk scenarios. By bridging semantic reasoning, transparency, and safety assurance, VLMs complement traditional perception systems and pave the way for more reliable and interpretable human-centric mobility solutions.

CHAPTER 6

Uncertainty Quantification through Conformal Prediction

6.1 Introduction

This chapter further extends the perception framework of Chapter 3 by embedding principled uncertainty quantification. Specifically, Conformal Prediction (CP) is applied to object detection pipelines to construct rigorous distribution-free confidence bounds around predictions. By complementing the deterministic outputs of YOLOv8 with calibrated uncertainty estimates, this stage enhances the reliability of perception in safety-critical domains, bridging the gap between raw detection accuracy and statistically guaranteed trustworthiness.

6.1.1 Background and Motivation

Autonomous Driving Systems (ADS) and Advanced Driver-Assistance Systems (ADAS) operate in safety-critical environments where perception models must not only detect and classify objects accurately but also provide reliable estimates of their uncertainty. Without principled Uncertainty Quantification (UQ), autonomous systems risk making overconfident yet incorrect decisions, potentially leading to hazardous outcomes such as collisions or unsafe maneuvers. The same concern applies broadly to other domains such as mobile robotics and medical imaging, where reliability and robustness are non-negotiable.

Deep Neural Networks (DNNs), despite their remarkable predictive accuracy, are known to produce miscalibrated confidence estimates, particularly in out-of-distribution or ambiguous scenarios [96]. In object detection—a core task for ADS perception pipelines—such miscalibration can manifest as bounding boxes with high confidence but incorrect localization or class labels. This discrepancy between reported confidence and true correctness undermines trust and complicates downstream decision-making processes.

Traditional methods for UQ attempt to address this gap. Bayesian neural networks [88] explicitly model weight uncertainty but are computationally intensive and difficult

to scale to modern vision tasks. Deep ensembles [169] improve robustness by averaging predictions across multiple models but require significant training overhead. Post-hoc calibration methods such as Platt scaling or temperature scaling [95] are simple and effective in certain cases but lack finite-sample guarantees and often fail under distribution shifts. As a result, none of these approaches fully resolve the need for principled, efficient, and generalizable UQ in safety-critical perception.

A promising solution to these limitations is Conformal Prediction (CP), originally introduced by Vovk, Shafer, and Gammerman [170, 171]. Unlike traditional calibration techniques, CP provides formal coverage guarantees under minimal assumptions. Its potential for efficient, post-hoc UQ in black-box models motivates a closer examination, which is outlined in the following subsection.

6.1.2 Conformal Prediction

Conformal Prediction (CP) is a distribution-free and model-agnostic framework for Uncertainty Quantification (UQ). Given a user-specified error rate α, CP augments the output of a predictor with a set or interval that is guaranteed to contain the true outcome with high probability. These guarantees are valid in finite samples and hold under the mild assumption of data exchangeability. Importantly, CP operates as a post-hoc wrapper: it requires no retraining of the underlying model and incurs only modest computational cost, making it attractive for real-time applications such as autonomous driving and other safety-critical perception systems.

6.1.2.1 In Image Classification

The first applications of CP in computer vision appeared in image classification. The key idea is to produce *prediction sets* that adapt to the difficulty of the input. For clear and unambiguous images, the set typically contains a single label, resembling the output of a conventional classifier. For more difficult or ambiguous cases, however, the prediction set expands to include multiple plausible labels. This adaptivity explicitly communicates predictive uncertainty, transforming point predictions into transparent, human-interpretable outputs that are directly useful in safety-critical decision-making.

Figure 6.1 illustrates this principle on ImageNet data. For easy cases, CP outputs a single label, while for ambiguous cases the set grows, reflecting uncertainty and reducing the risk of overconfident mistakes.



Figure 6.1: Conformal Prediction (CP) in image classification: prediction sets shrink for easy inputs and expand for ambiguous ones.

6.1.2.2 In Object Detection

Extending CP to object detection is significantly more challenging. Unlike classification, which produces a single label, object detection must predict both class labels and bounding boxes for a variable number of objects per image. Uncertainty therefore arises not only from classification ambiguity but also from localization errors, False Positive (FP), and missed detections. Addressing these complexities requires specialized conformal strategies.

Recent works have proposed conformalized bounding boxes [99, 3, 2], which enlarge predicted boxes using margins estimated from calibration data. This approach, known as box-wise conformalization, ensures that the ground-truth object is contained within the conformalized bounding box with high probability. An alternative line of work, Conformal Risk Control (CRC) [103], generalizes CP by allowing guarantees to be defined in terms of more flexible loss functions, such as bounding-box recall or pixel coverage.

Figure 6.2 illustrates the concept: red boxes denote ground-truth annotations, while shaded regions show conformalized bounding boxes. Tighter intervals correspond to confident detections, whereas wider intervals indicate higher uncertainty and can be interpreted as safety margins in downstream planning.

From the perspective of explainable and trustworthy Artificial Intelligence (AI), CP enhances transparency by providing formal, user-specified confidence guarantees—an effect that can be described as *explainability-through-certification*.

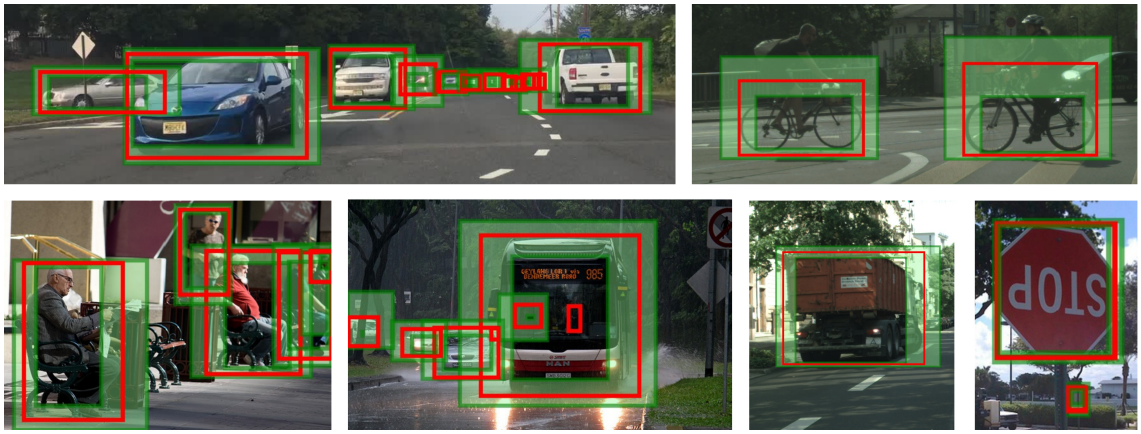


Figure 6.2: CP intervals for object detection. Red boxes denote ground-truth annotations, while shaded regions show conformalized bounding boxes with probabilistic coverage guarantees [2].

6.1.3 Contributions

This chapter focuses on providing statistical guarantees for the outputs of object detectors in safety-critical scenarios, using state-of-the-art CP methodologies. The main contributions are:

1. Application of CP to a custom object detector and task-specific categories relevant for assistive and indoor autonomous mobility.

2. Systematic comparison of multiple conformalization strategies, including box-wise approaches (Additive, Multiplicative, Bonferroni, and max-residual margins).

3. Implementation and evaluation of image-wise CP and Conformal Risk Control (CRC), using losses based on Hausdorff distance, box recall, and pixel recall, and identification of their practical limitations in this context.

4. Comparative analysis of object-level and scene-level UQ methods within the CP framework, providing insights into their practical applicability for safety-critical detection tasks.

6.2 Methodology

6.2.1 Conformal Prediction Fundamentals

Conformal Prediction (CP) provides a principled framework to construct prediction sets with finite-sample coverage guarantees. Consider a dataset $(X_1, Y_1), \ldots, (X_n, Y_n)$ drawn exchangeably from an unknown distribution P_{XY}, and let $\hat{f}$ denote a trained predictor. The objective is to produce, for a new input X, a set $C_\alpha(X)$ such that

$$\mathbb{P}\big(Y \in C_\alpha(X)\big) \geq 1 - \alpha, \tag{6.1}$$

where $\alpha \in (0, 1)$ is a user-specified error tolerance. This guarantee holds for any underlying predictor and requires only the mild assumption of data exchangeability, making CP particularly attractive for safety-critical domains.

The construction of $C_\alpha(X)$ relies on a nonconformity score $s(X, Y)$, which measures how unusual the candidate pair (X, Y) is relative to the model's predictions. A dedicated *calibration set* D_{cal} is used to compute such scores, which are then ranked to determine the $(1 - \alpha)$ empirical quantile. At inference, the prediction set is defined as

$$C_\alpha(X) = \{y : s(X, y) \leq q_{1-\alpha}\}.$$

In this way, the procedure guarantees that the true outcome is contained in $C_\alpha(X)$ with probability at least $1 - \alpha$.

In practice, a *split conformal prediction* scheme is adopted. The available data is divided into three disjoint parts: (1) a training set D_{train} to fit the base predictor $\hat{f}$, (2) a calibration set D_{cal} to compute nonconformity scores and estimate the conformal quantile, and (3) a test set D_{test} to evaluate empirical coverage and efficiency. This design makes CP computationally efficient, since it operates post-hoc and requires no retraining of the base model.

Algorithm 1 summarizes the split CP procedure.

Algorithm 1: Split Conformal Prediction (general procedure)

Input : Training set D_{train}, calibration set D_{cal}, miscoverage level α, nonconformity score $s(X, Y)$.

Output Prediction set $C_\alpha(X)$ with guaranteed coverage.

:

1. Train the base predictor $\hat{f}$ on D_{train}.
2. For each $(X_i, Y_i) \in D_{\text{cal}}$, compute the nonconformity score $R_i = s(X_i, Y_i)$.
3. Compute the conformal quantile $q_{1-\alpha}$ as the $(1-\alpha)$ empirical quantile of $\{R_i\}$.
4. For a new input X, define the prediction set:

$$C_\alpha(X) = \{y : s(X, y) \leq q_{1-\alpha}\}.$$

In image classification, nonconformity scores are often derived from prediction errors or softmax confidences, leading to set-valued outputs that adapt to input difficulty. For unambiguous images, the prediction set may reduce to a single label, whereas for ambiguous inputs, it expands to include multiple plausible classes. This illustrates how CP transforms conventional model outputs into rigorous, uncertainty-aware predictions.

Extending this principle to object detection introduces additional complexity. Unlike classification, which requires only a single label, detection involves predicting a variable number of bounding boxes and class labels. Uncertainty may therefore arise from both classification errors and localization inaccuracies, as well as False Positives (FPs) and missed detections. To address these challenges, specialized conformalization strategies have been developed, which are presented in the following subsections.

6.2.2 Box-wise Conformal Prediction

The most direct way to extend CP to object detection is through *box-wise conformalization*. In this setting, each predicted bounding box is treated individually, and residuals between predicted and ground-truth boxes are used to calibrate margins. The outcome is a conformalized bounding box that contains the true object with probability at least $1 - \alpha$.

6.2.2.1 Bounding box representation and matching

Let $Y^k = (x^k_{\min}, y^k_{\min}, x^k_{\max}, y^k_{\max})$ denote the ground-truth coordinates of the k-th object and $\hat{Y}^k = (\hat{x}^k_{\min}, \hat{y}^k_{\min}, \hat{x}^k_{\max}, \hat{y}^k_{\max})$ the corresponding detector prediction. Bounding boxes

may equivalently be defined as the set of pixels within these coordinates:

$$B^k = \{(x, y) \in \mathbb{R}^2 : x \in [x^k_{\min}, x^k_{\max}],\ y \in [y^k_{\min}, y^k_{\max}]\}.$$

Since object detectors output a variable set of boxes, a pairing between predicted and ground-truth boxes is required. Following prior work [99, 3], Hungarian matching with an Intersection-over-Union (IoU) threshold is applied to establish correspondences. The conformal guarantee therefore applies exclusively to True Positive (TP)s, i.e., matched predictions. The choice of IoU threshold is important: a stricter threshold results in fewer matches but tighter guarantees, whereas a looser threshold yields more matches but potentially larger margins.

6.2.2.2 Nonconformity scores

Once matched pairs are established, the difference between predicted and ground-truth coordinates is used to define residuals, which serve as nonconformity scores. Several formulations are possible.

Additive score. Residuals are computed as fixed pixel differences:

$$R^k = \left[\hat{x}^k_{\min} - x^k_{\min},\ \hat{y}^k_{\min} - y^k_{\min},\ x^k_{\max} - \hat{x}^k_{\max},\ y^k_{\max} - \hat{y}^k_{\max}\right]. \tag{6.2}$$

In Additive conformalization, margins are expressed in pixels and remain independent of object size.

Multiplicative score. To account for object scale, residuals may be normalized by the predicted width b^k_w and height b^k_h:

$$R^k = \left[\frac{\hat{x}^k_{\min} - x^k_{\min}}{b^k_w},\ \frac{\hat{y}^k_{\min} - y^k_{\min}}{b^k_h},\ \frac{x^k_{\max} - \hat{x}^k_{\max}}{b^k_w},\ \frac{y^k_{\max} - \hat{y}^k_{\max}}{b^k_h}\right]. \tag{6.3}$$

In Multiplicative conformalization, margins scale proportionally to the predicted width and height of the object.

Figure 6.3 illustrates box-wise conformalization through Additive and Multiplicative nonconformity scores.

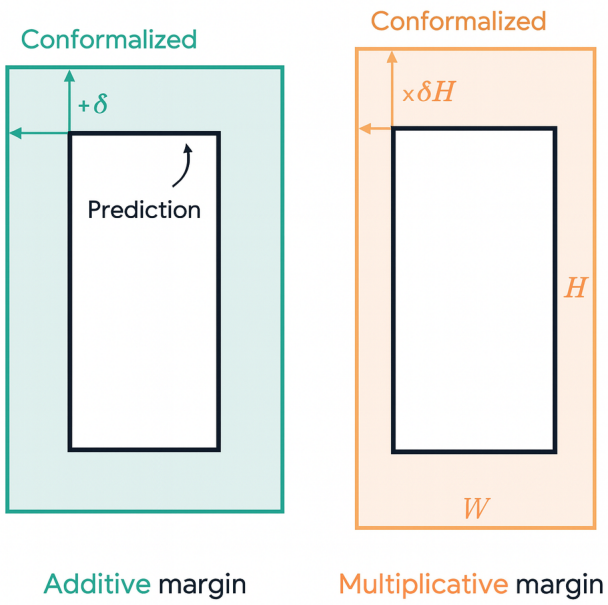


Figure 6.3: Box-wise conformalization illustrated through Additive and Multiplicative nonconformity scores [3].

Bonferroni correction. Bounding boxes consist of four coordinates, and each coordinate can fail independently. To guarantee joint coverage, a multiple-testing adjustment is required. Following [99], a Bonferroni correction assigns a miscoverage rate of $\alpha/4$ to each coordinate. While statistically valid, this procedure is often overly conservative and may lead to inflated margins, as illustrated in Figure 6.4.

Max-Additive and Max-Multiplicative scores. To reduce conservativeness, [3] proposed using the maximum residual across coordinates. The Max-Additive score is

$$R^k_{\max} = \max\left(\hat{x}^k_{\min} - x^k_{\min},\ \hat{y}^k_{\min} - y^k_{\min},\ x^k_{\max} - \hat{x}^k_{\max},\ y^k_{\max} - \hat{y}^k_{\max}\right), \tag{6.4}$$

while the Max-Multiplicative score is

$$R^k_{\max} = \max\left(\frac{\hat{x}^k_{\min} - x^k_{\min}}{b^k_w},\ \frac{\hat{y}^k_{\min} - y^k_{\min}}{b^k_h},\ \frac{x^k_{\max} - \hat{x}^k_{\max}}{b^k_w},\ \frac{y^k_{\max} - \hat{y}^k_{\max}}{b^k_h}\right). \tag{6.5}$$

Equations 6.4 and 6.5 reduce calibration to a single quantile, typically producing tighter and less conservative margins than Bonferroni correction. This difference is illustrated in Figure 6.4.

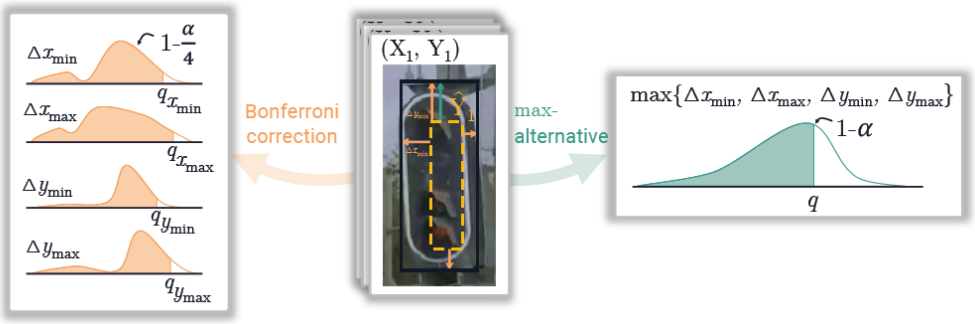


Figure 6.4: Comparison of Bonferroni correction and max-residual conformalization. Bonferroni estimates four coordinate-wise quantiles at level $\alpha/4$, leading to conservative margins, whereas the max-residual method uses a single quantile at level α, yielding tighter bounds [3].

6.2.2.3 Conformal quantile and prediction set

Let $\{R^k\}_{k=1}^n$ denote the residuals obtained from the calibration set. Sorting these values, the conformal quantile is defined as

$$q_{1-\alpha} = R^{(\lceil (n+1)(1-\alpha) \rceil)}.$$

Intuitively, $q_{1-\alpha}$ is the margin large enough to cover at least $1-\alpha$ fraction of calibration residuals.

During inference, for a new predicted box $\hat{Y} = (\hat{x}_{\min}, \hat{y}_{\min}, \hat{x}_{\max}, \hat{y}_{\max})$, the conformalized box is obtained as

$$\hat{C}_{1-\alpha}(X) = (\hat{x}_{\min} - \Delta_x,\ \hat{y}_{\min} - \Delta_y,\ \hat{x}_{\max} + \Delta_x,\ \hat{y}_{\max} + \Delta_y), \tag{6.6}$$

where Δ_x, Δ_y correspond to quantiles derived from the chosen nonconformity score (Additive, Multiplicative, Max-Additive, or Max-Multiplicative).

In Summary, box-wise conformalization provides object-level UQ by enlarging predicted bounding boxes with calibrated margins. The guarantees apply to matched detections and ensure that, with probability at least $1-\alpha$, the conformalized box contains the

true object. Among the various formulations, max-residual approaches generally yield less conservative but effective solutions, making them particularly appealing in practical applications.

6.2.3 Image-wise Conformal Prediction and Conformal Risk Control

While box-wise conformalization provides guarantees for individual detections, it requires pairing predicted and ground-truth boxes and therefore applies only to TPs. This leaves open the possibility of missed detections (False Negative (FN)s), which are critical in safety applications. To address this limitation, *image-wise conformalization* extends the conformal framework to entire scenes. In this setting, the ground truth Y_i is the set of bounding boxes in image X_i, and nonconformity scores (or losses) are computed over all objects in the image. This approach avoids explicit matching and focuses on the proportion of ground-truth objects that are not adequately covered.

6.2.3.1 Hausdorff-based conformalization

An early approach, proposed by [99], uses an asymmetric Hausdorff distance. For each image, the score is the smallest margin that must be applied uniformly to all predicted boxes so that at least a fraction $1 - \phi$ of ground-truth objects are covered. In practice, ϕ is typically set to 0.25, i.e., a 75% coverage target. Although simple, this criterion is discontinuous: either the threshold proportion is met or it is not, which can cause abrupt changes in the required margin and lead to unstable behavior.

6.2.3.2 Conformal Risk Control (CRC)

To overcome these limitations, [103] introduced Conformal Risk Control (CRC), which replaces binary miscoverage with a bounded loss $\ell(\cdot)$ and guarantees that the expected loss does not exceed a user-specified level α. Let $C_\tau(X)$ denote the set of *conformalized predictions* obtained by enlarging all detector boxes in image X with a margin parameter τ (e.g., Additive or Multiplicative). Given calibration data $(X_i, Y_i)_{i=1}^n$, define

$$L_i(\tau) \;=\; \ell\big(C_\tau(X_i),\, Y_i\big),$$

and select $\hat{\tau}$ so that the (empirical) risk constraint is satisfied. The target guarantee is

$$\mathbb{E}[\, L(\hat{\tau})\,] \;\le\; \alpha,$$

i.e., the expected proportion of missed content is at most α.

6.2.3.3 Loss functions for object detection

Two recall-oriented losses are commonly used to evaluate coverage after enlargement by τ. Let n_i denote the number of ground-truth boxes in X_i, and let $\widehat{C}_\tau(X_i) = \bigcup_j \hat{C}_\tau(\hat{Y}_i^j)$ be the union of all *conformalized* predicted boxes in image X_i.

Box-wise recall loss.

$$L_i^{\text{box}}(\tau) = \begin{cases} 0, & n_i = 0, \\ 1 - \dfrac{1}{n_i} \displaystyle\sum_{k=1}^{n_i} \mathbf{1}\left\{ Y_i^k \subseteq \widehat{C}_\tau(X_i) \right\}, & \text{otherwise.} \end{cases} \tag{6.7}$$

This measures the fraction of ground-truth boxes not fully covered by conformalized predictions. Multiple predictions may jointly cover a single ground-truth box.

Pixel-wise recall loss.

$$L_i^{\text{pixel}}(\tau) = \begin{cases} 0, & n_i = 0, \\ 1 - \dfrac{1}{n_i} \displaystyle\sum_{k=1}^{n_i} \dfrac{A\left(Y_i^k \cap \widehat{C}_\tau(X_i)\right)}{A\left(Y_i^k\right)}, & \text{otherwise,} \end{cases} \tag{6.8}$$

where $A(\cdot)$ denotes pixel area. This loss is smoother than the box-wise recall and implicitly weights errors by object size, penalizing missed large objects more than small ones. (Neither loss directly penalizes FPs; they focus on recall.)

6.2.3.4 Image-wise CRC algorithm

The CRC procedure follows the split-conformal paradigm, using a calibration set to tune τ. Because $L_i(\tau)$ is non-increasing in τ (larger margins cannot reduce coverage), $\hat{\tau}$ can be found by a line or binary search. The complete procedure is summarized in Algorithm 2.

Algorithm 2: Image-wise Conformal Risk Control for Object Detection

Input : Training set D_{train}, calibration set D_{cal}, risk level α, loss $\ell(\cdot)$, search grid $\mathcal{T}$, margin rule (Additive or Multiplicative).

Output : Conformalized predictions $\hat{C}_{\hat{\tau}}(X)$ with image-level risk control.

1. Train the detector $\hat{f}$ on D_{train}.
2. For each $\tau \in \mathcal{T}$ and each $(X_i, Y_i) \in D_{\text{cal}}$, compute $L_i(\tau) = \ell\big(C_\tau(X_i), Y_i\big)$.
3. Compute the empirical risk $\hat{R}(\tau) = \frac{1}{|D_{\text{cal}}|} \sum_i L_i(\tau)$.
4. Select $\hat{\tau} = \min\{\tau \in \mathcal{T} : \hat{R}(\tau) \leq \alpha\}$ (binary search optional).
5. At inference, enlarge each predicted box in a new image X using $\hat{\tau}$ and the chosen margin rule to form $\hat{C}_{\hat{\tau}}(X)$.

6.2.3.5 Conformalized prediction sets

At inference, conformalized boxes are constructed exactly as in the box-wise case, but with the image-level margin $\hat{\tau}$. For the Additive rule,

$$\hat{C}_{\hat{\tau}}(X) \;=\; (\hat{x}_{\min} - \hat{\tau},\ \hat{y}_{\min} - \hat{\tau},\ \hat{x}_{\max} + \hat{\tau},\ \hat{y}_{\max} + \hat{\tau}), \tag{6.9}$$

and for the Multiplicative rule,

$$\hat{C}_{\hat{\tau}}(X) \;=\; (\hat{x}_{\min} - b_w\hat{\tau},\ \hat{y}_{\min} - b_h\hat{\tau},\ \hat{x}_{\max} + b_w\hat{\tau},\ \hat{y}_{\max} + b_h\hat{\tau}), \tag{6.10}$$

where b_w and b_h are the predicted box width and height.

In summary, image-wise conformalization provides scene-level guarantees by controlling the expected proportion of missed detections, rather than the coverage of individual boxes. The Hausdorff-based method offers a simple but discontinuous criterion, whereas CRC with recall-based losses yields smoother, tunable guarantees. These image-level approaches complement box-wise conformalization by quantifying uncertainty at a higher level of abstraction aligned with operational safety needs.

6.3 Experimental Validation

The experiments build upon the fine-tuned YOLOv8s detector introduced in Chapter 3. Two object categories were considered: *person* and *wheelchair*.

6.3.1 Datasets

Two datasets were used for CP experiments:

Calibration. A subset of 570 images was drawn from the validation set described in Chapter 3, Section 7.3.1, and employed to compute the conformal quantiles required for bounding-box calibration.

Test. Evaluation was performed on the CNR Villa benchmark, consisting of 714 images under realistic and diverse conditions (see Chapter 3, Section 3.3.1.4). This dataset provides a rigorous testbed for UQ.

6.3.2 Evaluation Metrics

Four complementary metrics were employed.

Coverage. The fraction of ground-truth boxes fully contained within predicted or conformalized boxes:

$$\text{Coverage} = \frac{1}{\sum_{i=1}^{N} |Y_i|} \sum_{i=1}^{N} \sum_{k=1}^{|Y_i|} \mathbf{1}\big(Y_i^k \subseteq \hat{C}_i^k\big), \tag{6.11}$$

where $|Y_i|$ is the number of ground-truth objects in image X_i.

Coverage (Pred) refers to raw YOLOv8s predictions, while Coverage (Conf) denotes post-conformalization results. Ideally, coverage approaches $1 - \alpha$.

Risk. Defined as the fraction of ground-truth boxes or pixels not adequately covered, i.e.,

$$\text{Risk} = 1 - \text{Coverage}.$$

Risk is particularly relevant for image-wise conformalization, where it quantifies scene-level failures.

Stretch. The multiplicative increase in bounding-box area due to conformalization:

$$\text{Stretch} = \frac{1}{\sum_{i=1}^{N} |\hat{Y}_i|} \sum_{i=1}^{N} \sum_{k=1}^{|\hat{Y}_i|} \frac{A(\hat{C}_i^k)}{A(\hat{Y}_i^k)}, \tag{6.12}$$

where $A(\cdot)$ denotes box area. Values close to 1 indicate efficient expansions.

Margins. The calibrated offsets applied to box boundaries. In the additive case:

$$\hat{C}^k = \big(\hat{x}_{\min} - q_{1-\alpha},\ \hat{y}_{\min} - q_{1-\alpha},\ \hat{x}_{\max} + q_{1-\alpha},\ \hat{y}_{\max} + q_{1-\alpha}\big), \tag{6.13}$$

and in the multiplicative case:

$$\hat{C}^k = \big(\hat{x}_{\min} - b_w q_{1-\alpha},\ \hat{y}_{\min} - b_h q_{1-\alpha},\ \hat{x}_{\max} + b_w q_{1-\alpha},\ \hat{y}_{\max} + b_h q_{1-\alpha}\big). \tag{6.14}$$

Margins quantify the adjustments required to guarantee $1 - \alpha$ coverage.

6.3.3 Results

6.3.3.1 Box-wise Results at Default $\alpha = 0.1$

Four methods were evaluated: Max-Additive, Max-Multiplicative, Additive with Bonferroni correction, and Multiplicative with Bonferroni correction. Results are reported separately for calibration and test sets in Tables 6.1 and 6.2.

On both datasets, conformalization substantially increased coverage relative to baseline YOLOv8s (0.78 calibration, 0.88 test). Max-Additive achieved the most compact expansions, with stretch around 1.25. Max-Multiplicative delivered similar coverage but required larger boxes (stretch $\approx$ 1.6–1.9). Bonferroni correction moderated this effect: both additive and multiplicative corrections reduced stretch while retaining reliable coverage.

Table 6.1: Box-wise conformalization results on the calibration set at $\alpha = 0.1$.

Method	Coverage (Pred)	Coverage (Conf)	Stretch	Margin(s)
Max-Additive	0.78	0.97	1.266	0.0796
Additive + Bonferroni	0.78	0.97	1.251	[0.092, 0.075, 0.085, 0.044]
Max-Multiplicative	0.78	0.95	1.874	0.8611
Multiplicative + Bonferroni	0.78	0.95	1.510	[1.324, 0.199, 1.252, 0.180]

Table 6.2: Box-wise conformalization results on the test set at $\alpha = 0.1$.

Method	Coverage (Pred)	Coverage (Conf)	Stretch	Margin(s)
Max-Additive	0.88	0.96	1.283	0.0796
Additive + Bonferroni	0.88	0.95	1.267	[0.092, 0.075, 0.085, 0.044]
Max-Multiplicative	0.88	0.94	1.640	0.8611
Multiplicative + Bonferroni	0.88	0.92	1.344	[1.324, 0.199, 1.252, 0.180]

These results confirm the calibration effect of CP: coverage rises into the 0.92–0.97 range, at the cost of expanded box size. The balance between reliability and efficiency is

most favorable for Max-Additive, while Bonferroni corrections provide robustness across settings.

Figure 6.5 illustrates representative detections for the four methods. Max-Additive yields compact expansions, while Max-Multiplicative results in visibly larger boxes. Bonferroni corrections mitigate this inflation, producing more efficient intervals while preserving coverage.



(a) Max-Additive (b) Additive with Bonferroni Correction

(c) Max-Multiplicative (d) Multiplicative with Bonferroni Correction

Figure 6.5: Representative box-wise conformalization results at $\alpha = 0.1$: (a) Max-Additive, (b) Additive with Bonferroni correction, (c) Max-Multiplicative, (d) Multiplicative with Bonferroni correction. Purple: conformalized predictions; yellow: YOLOv8s prediction; blue: ground truth.

6.3.3.2 Effect of α on Box-wise Conformalization

The parameter α determines the nominal miscoverage rate and thus the trade-off between reliability and efficiency. Experiments were conducted for $\alpha \in \{0.01, 0.02, 0.05, 0.1, 0.15, 0.2, 0.3\}$ across all four methods, with results reported in Tables 6.3–6.4.

Across methods, conformalization improved coverage compared to baseline YOLOv8s (0.78 calibration, 0.88 test). Max-Additive maintained near-nominal coverage (0.95–0.99)

Table 6.3: Coverage, stretch, and margins for four box-wise conformalization methods across α values (Calibration set).

Method	α	Coverage (Pred)	Coverage (Conf)	Stretch	Margin(s)
Max-Additive	0.01	0.78	0.97	1.540	0.153
	0.05	0.78	0.97	1.330	0.097
	0.10	0.78	0.97	1.266	0.080
	0.20	0.78	0.98	1.158	0.048
	0.30	0.78	0.97	1.066	0.021
Additive + Bonferroni	0.01	0.78	0.97	1.713	[0.138, 0.307, 0.153, 0.210]
	0.05	0.78	0.96	1.355	[0.107, 0.134, 0.097, 0.079]
	0.10	0.78	0.97	1.251	[0.092, 0.075, 0.085, 0.044]
	0.20	0.78	0.97	1.173	[0.076, 0.039, 0.065, 0.025]
	0.30	0.78	0.97	1.118	[0.063, 0.025, 0.034, 0.018]
Max-Multiplicative	0.01	0.78	0.98	3.757	2.181
	0.05	0.78	0.96	2.364	1.252
	0.10	0.78	0.95	1.874	0.861
	0.20	0.78	0.97	1.326	0.355
	0.30	0.78	0.97	1.056	0.065
Multiplicative + Bonferroni	0.01	0.78	0.98	3.510	[3.381, 1.145, 4.554, 0.989]
	0.05	0.78	0.96	1.746	[1.811, 0.356, 1.534, 0.283]
	0.10	0.78	0.95	1.510	[1.324, 0.199, 1.252, 0.180]
	0.20	0.78	0.95	1.302	[0.918, 0.114, 0.777, 0.070]
	0.30	0.78	0.96	1.199	[0.749, 0.071, 0.421, 0.041]

Table 6.4: Coverage, stretch, and margins for four box-wise conformalization methods across α values (Test set).

Method	α	Coverage (Pred)	Coverage (Conf)	Stretch	Margin(s)
Max-Additive	0.01	0.88	0.96	1.576	0.153
	0.05	0.88	0.96	1.351	0.097
	0.10	0.88	0.96	1.283	0.080
	0.20	0.88	0.97	1.168	0.048
	0.30	0.88	0.99	1.070	0.021
Additive + Bonferroni	0.01	0.88	0.96	1.762	[0.138, 0.307, 0.153, 0.210]
	0.05	0.88	0.95	1.378	[0.107, 0.134, 0.097, 0.079]
	0.10	0.88	0.95	1.267	[0.092, 0.075, 0.085, 0.044]
	0.20	0.88	0.96	1.183	[0.076, 0.039, 0.065, 0.025]
	0.30	0.88	0.97	1.125	[0.063, 0.025, 0.034, 0.018]
Max-Multiplicative	0.01	0.88	0.95	2.890	2.181
	0.05	0.88	0.95	1.977	1.252
	0.10	0.88	0.94	1.640	0.861
	0.20	0.88	0.96	1.247	0.355
	0.30	0.88	0.99	1.043	0.065
Multiplicative + Bonferroni	0.01	0.88	0.96	2.619	[3.381, 1.145, 4.554, 0.989]
	0.05	0.88	0.92	1.502	[1.811, 0.356, 1.534, 0.283]
	0.10	0.88	0.92	1.344	[1.324, 0.199, 1.252, 0.180]
	0.20	0.88	0.94	1.203	[0.918, 0.114, 0.777, 0.070]
	0.30	0.88	0.96	1.134	[0.749, 0.071, 0.421, 0.041]

with consistently low stretch (1.07–1.54). Max-Multiplicative also achieved high coverage but required inflated boxes at low α (stretch up to 3.76 calibration, 2.89 test), tightening as α increased. Bonferroni corrections added conservatism: Additive with Bonferroni

correction incurred slightly higher stretch than Max-Additive, while Multiplicative with Bonferroni correction significantly reduced the inflation seen in Max-Multiplicative.

Figures 6.6 and 6.7 (test set) summarize these trends. Coverage consistently approaches or exceeds the nominal target with increasing α, while stretch demonstrates the efficiency trade-off: additive methods remain compact, and multiplicative methods shrink from highly inflated to near-additive levels. Figure 6.8 provides visual confirmation, illustrating the impact of $\alpha = 0.01$, 0.15, and 0.3.

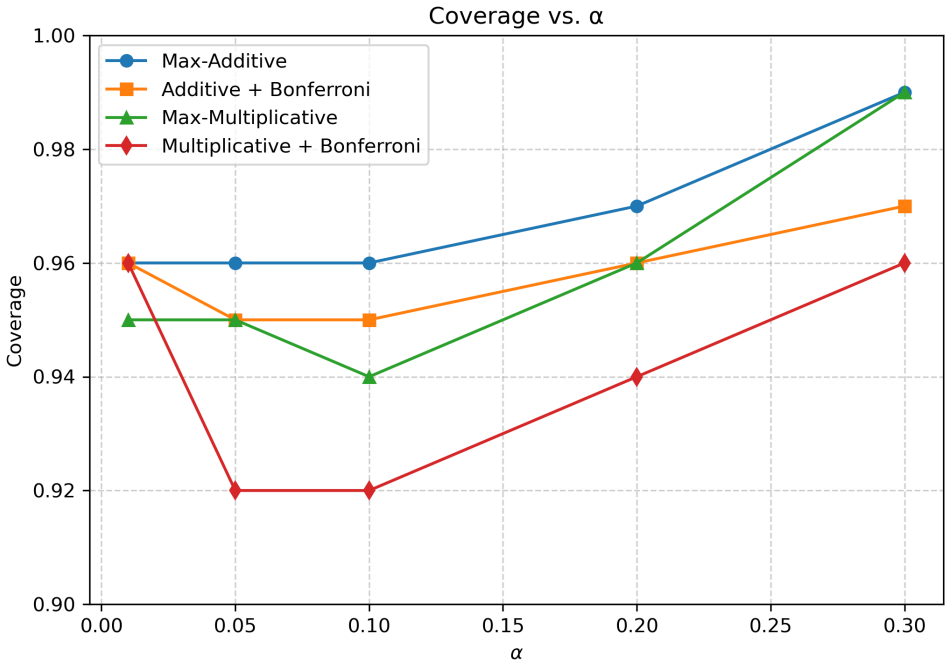


Figure 6.6: Coverage as a function of α for four box-wise conformalization methods on the test set. Max-Additive achieves consistently high coverage; Multiplicative with Bonferroni correction lags at low α but recovers at higher values.

In addition to the quantitative results, Figure 6.8 illustrates how varying α affects bounding-box conformalization. Representative examples are shown for $\alpha = 0.01$, 0.15, and 0.3 across the four methods. At $\alpha = 0.01$, margins are highly inflated—especially for the multiplicative methods—reflecting conservative guarantees. At $\alpha = 0.15$, boxes become visibly tighter while retaining coverage. At $\alpha = 0.3$, boxes are compact, with Max-Additive and Additive with Bonferroni correction producing the most efficient intervals.

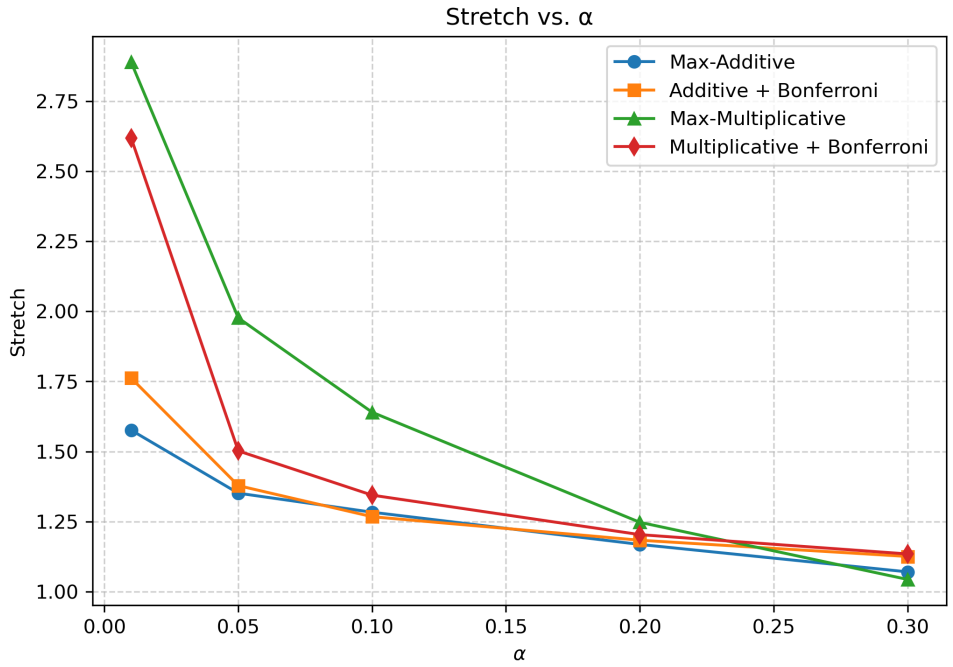


Figure 6.7: Stretch as a function of α for four box-wise conformalization methods on the test set. Max-Additive remains most efficient across α, while Max-Multiplicative starts highly inflated but converges toward additive levels as α increases.

6.3.3.3 Image-wise Results at Default $\alpha = 0.1$

Image-wise conformalization extends uncertainty quantification from individual detections to entire scenes. Three loss formulations were evaluated: Hausdorff distance, box recall loss, and pixel recall loss, each with Additive and Multiplicative margins. Results are reported for calibration and test sets in Tables 6.5–6.6.

Across all methods, baseline YOLOv8s predictions had risk = 1.00. Conformalization reduced risk substantially: Additive margins lowered risk to ≈ 0.18, while Multiplicative margins reduced it only to 0.54–0.55. This improvement came at the cost of stretch, which was ≈ 3.67 for Additive and ≈ 3.53 for Multiplicative strategies. All three loss functions produced nearly identical outcomes, indicating that margin type is more decisive than the specific loss. Due to the high stretch, conformalized boxes often exceeded image boundaries, limiting the usefulness of qualitative visualization.

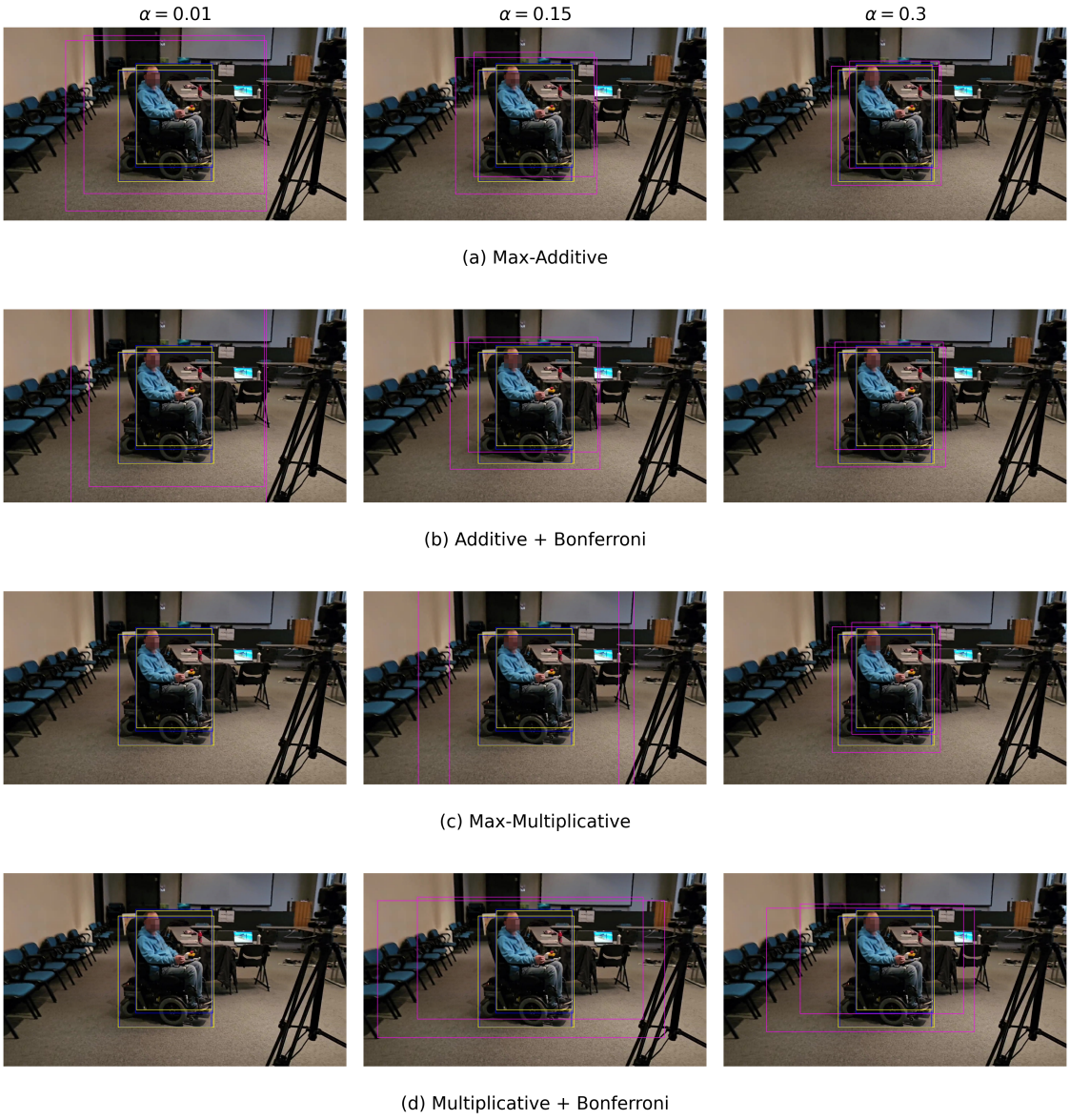


Figure 6.8: Representative conformalized predictions for $\alpha = 0.01$, 0.15, and 0.3 on the test set. Rows correspond to methods: (a) Max-Additive, (b) Additive with Bonferroni correction, (c) Max-Multiplicative, (d) Multiplicative with Bonferroni correction. Purple: conformalized predictions; yellow: YOLOv8s predictions; blue: ground truth.

Table 6.5: Calibration set results at $\alpha = 0.1$ for image-wise conformalization.

Loss Function	Margin Type	Risk (Pred)	Risk (Conf)	Stretch
Hausdorff Distance	Additive	1.00	0.18	3.669
	Multiplicative	1.00	0.55	3.527
Box Recall Loss	Additive	1.00	0.18	3.669
	Multiplicative	1.00	0.55	3.527
Pixel Recall Loss	Additive	1.00	0.18	3.669
	Multiplicative	1.00	0.54	3.527

Table 6.6: Test set results at $\alpha = 0.1$ for image-wise conformalization.

Loss Function	Margin Type	Risk (Pred)	Risk (Conf)	Stretch
Hausdorff Distance	Additive	1.00	0.18	3.669
	Multiplicative	1.00	0.55	3.527
Box Recall Loss	Additive	1.00	0.18	3.669
	Multiplicative	1.00	0.55	3.527
Pixel Recall Loss	Additive	1.00	0.18	3.669
	Multiplicative	1.00	0.54	3.527

6.3.3.4 Effect of α on Image-wise Conformalization

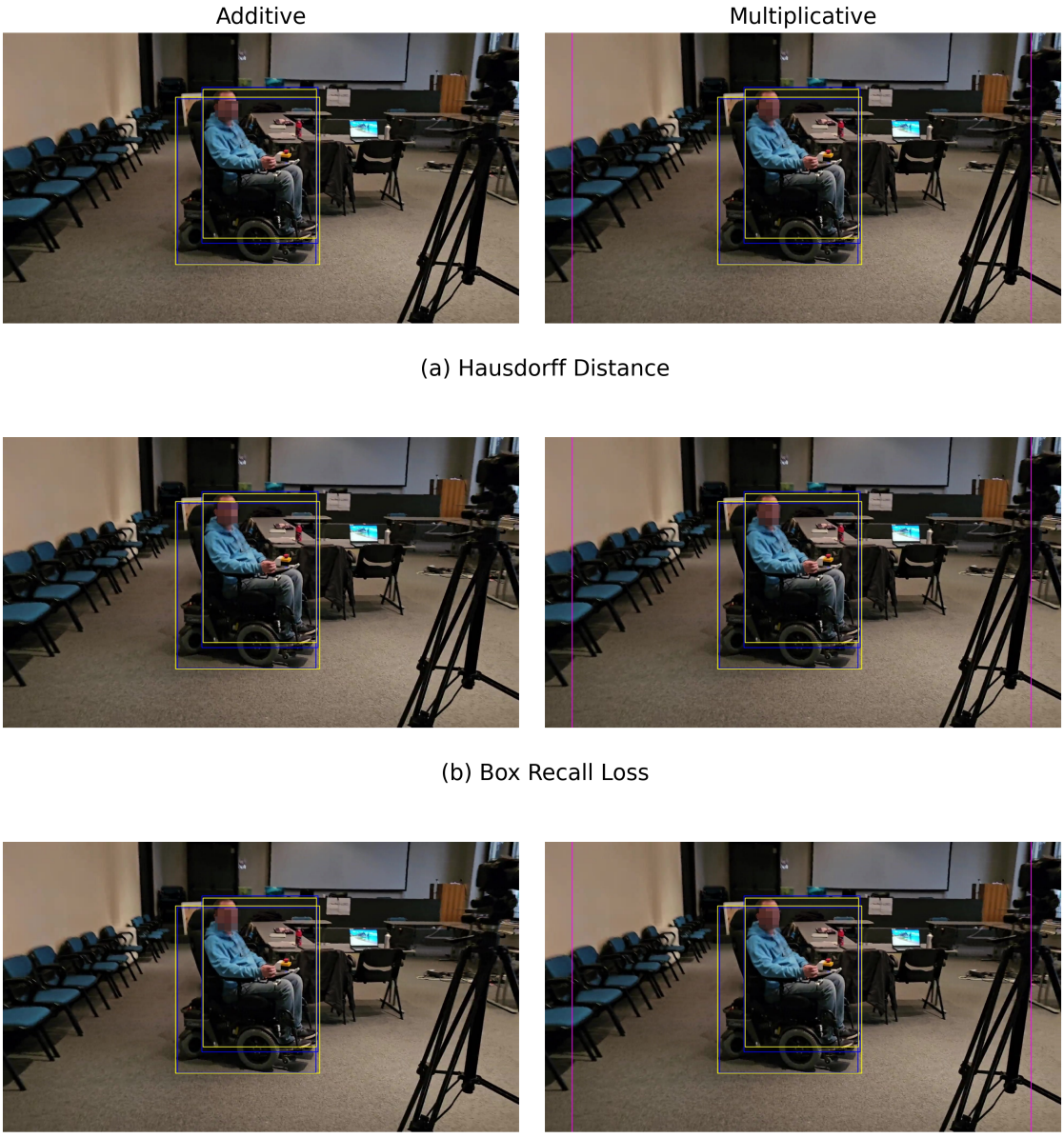
Additional experiments were performed at $\alpha = 0.3$ to examine the effect of relaxed guarantees (Tables 6.7–6.8).

Additive margins reduced risk to $\approx$ 0.45–0.47, but required stretch $\approx$ 2.93. Multiplicative margins yielded smaller expansions (stretch $\approx$ 2.23) but higher residual risk ($\approx$ 0.84). This highlights the trade-off: Additive emphasizes coverage robustness, while Multiplicative favors compact intervals at the expense of missed detections.

Figure 6.9 illustrates test-set examples at $\alpha = 0.3$. Rows correspond to Hausdorff, Box Recall, and Pixel Recall losses; columns compare Additive vs. Multiplicative margins. Additive strategies visibly inflate boxes, while Multiplicative leaves parts of the object uncovered.

6.3.3.5 Comparison of Box-wise and Image-wise Approaches

Box-wise conformalization provided reliable guarantees at the object level, raising coverage from baseline levels (0.78 calibration, 0.88 test) to 0.92–0.97, with stretch typically ≤ 1.5 for additive margins. In contrast, image-wise conformalization significantly reduced scene-level risk but incurred far larger expansions, with stretch exceeding 3.5 at $\alpha = 0.1$ and only partially improving at $\alpha = 0.3$.



(a) Hausdorff Distance

(b) Box Recall Loss

(c) Pixel Recall Loss

Figure 6.9: Representative results for $\alpha = 0.3$ on the test set. Rows: (a) Hausdorff Distance, (b) Box Recall Loss, (c) Pixel Recall Loss. Columns: Additive vs. Multiplicative. Purple: conformalized; yellow: YOLOv8s; blue: ground truth.

Table 6.7: Calibration set results at $\alpha = 0.3$ for image-wise conformalization.

Loss Function	Margin Type	Risk (Pred)	Risk (Conf)	Stretch
Hausdorff Distance	Additive	1.00	0.47	2.932
	Multiplicative	1.00	0.84	2.229
Box Recall Loss	Additive	1.00	0.47	2.932
	Multiplicative	1.00	0.84	2.229
Pixel Recall Loss	Additive	1.00	0.45	2.932
	Multiplicative	1.00	0.84	2.229

Table 6.8: Test set results at $\alpha = 0.3$ for image-wise conformalization.

Loss Function	Margin Type	Risk (Pred)	Risk (Conf)	Stretch
Hausdorff Distance	Additive	1.00	0.47	2.932
	Multiplicative	1.00	0.84	2.229
Box Recall Loss	Additive	1.00	0.47	2.932
	Multiplicative	1.00	0.84	2.229
Pixel Recall Loss	Additive	1.00	0.45	2.932
	Multiplicative	1.00	0.84	2.229

These results indicate that box-wise methods are practical for bounding-box calibration in safety-critical mobility tasks, where modest expansion is acceptable. Image-wise methods remain overly conservative, producing boxes that often exceed image boundaries.

6.4 Conclusion

This chapter investigated the role of Conformal Prediction (CP) as a framework for rigorous Uncertainty Quantification (UQ) in object detection, with a focus on indoor assistive mobility scenarios. Unlike conventional calibration approaches, CP provides finite-sample, distribution-free guarantees that are practically useful in safety-critical domains. The work systematically evaluated both box-wise and image-wise conformalization strategies, comparing their performance across calibration and test sets.

At the object level, box-wise conformalization demonstrated clear benefits. Across all methods, coverage improved substantially over the baseline YOLOv8s detector (0.78 calibration, 0.88 test), consistently reaching levels between 0.92 and 0.97. Among the tested variants, Max-Additive achieved the most favorable balance between reliability and efficiency, delivering near-nominal coverage with stretch values typically below 1.3. Max-Multiplicative attained similar coverage but at the cost of larger expansions, especially at

low α values. Bonferroni-corrected methods acted as an intermediate compromise: they curtailed the extreme inflation of Max-Multiplicative, though at the expense of slightly higher stretch compared to Max-Additive. These findings confirm that carefully selected conformalization strategies can reliably calibrate bounding boxes without incurring excessive efficiency costs.

The effect of the miscoverage parameter α further illustrated the inherent trade-off between robustness and compactness. Lower values of α (e.g., 0.01) led to overly inflated boxes, particularly for multiplicative methods, while higher values (e.g., 0.3) yielded compact and practical intervals. Coverage consistently approached or exceeded nominal targets as α increased, showing that α can serve as a user-controllable lever for tuning conservatism in deployment.

At the scene level, image-wise conformalization was evaluated through Hausdorff-based criteria and Conformal Risk Control (CRC) with recall-oriented losses. These methods successfully reduced risk from the baseline level of 1.0 to approximately 0.18 with Additive margins and to 0.54–0.55 with Multiplicative margins at $\alpha = 0.1$. However, these guarantees came with substantial box inflation: stretch values remained above 3.5, often extending boxes beyond image boundaries. Even at $\alpha = 0.3$, expansions were still excessive (≈ 2.9 for Additive, ≈ 2.2 for Multiplicative). The similarity of results across Hausdorff, box recall, and pixel recall losses indicates that margin type was more decisive than the specific loss formulation. Nevertheless, the scale of expansion observed limits the practical utility of current image-wise methods.

The comparison of object-level and scene-level strategies underscores a key distinction. Box-wise conformalization emerged as an effective and practical approach, producing compact intervals with reliable guarantees suitable for safety-critical perception. Image-wise methods, while conceptually appealing for scene-level risk control, were overly conservative under current formulations. This contrast highlights that while CP is versatile, its practical value depends on aligning the chosen strategy with the level of abstraction required and the acceptable trade-off between efficiency and conservatism.

Overall, the results demonstrate that CP provides a principled mechanism for introducing statistical guarantees into object detection pipelines. In particular, box-wise conformalization offers a viable solution for uncertainty quantification in assistive mobility contexts, combining strong reliability with manageable efficiency costs. Image-wise approaches, although theoretically stronger, remain too conservative for deployment in their current form. Together, these findings establish CP as a valuable tool for building trustworthy, uncertainty-aware perception systems in safety-critical domains.

CHAPTER 7

Explainability in Federated Learning Systems

7.1 Introduction

This chapter investigates the integration of explainability into Federated Learning (FL) systems, extending the broader theme of this thesis from video analytics to distributed machine learning. While the preceding chapters have focused on perception, monitoring, and rule-based verification in vision-centric domains, the current discussion shifts toward distributed learning paradigms where data privacy, ownership, and interpretability play a central role.

7.1.1 Scope and Motivation

In the rapidly evolving landscape of AI applications, the reliance of Machine Learning (ML) methods on large-scale training datasets has become a critical challenge. Increasingly, data is fragmented across multiple organizations or devices, creating "data islands" that cannot be easily centralized. Traditional approaches to ML, which assume access to a centralized dataset, require extensive data transfer to a single coordination point. Such practices introduce high communication costs and raise concerns regarding data ownership [172], privacy [173, 174], and security [175], particularly under strict regulatory frameworks such as the General Data Protection Regulation (GDPR) [31].

To address these limitations, Federated Learning (FL) [32] has emerged as a transformative paradigm. FL enables collaborative model training across distributed clients without requiring data to leave its local environment, thereby preserving privacy and ownership. This paradigm has gained traction in domains such as Internet-of-Things (IoT), healthcare, and edge computing, where data locality and regulatory compliance are critical. By avoiding data centralization, FL mitigates the risks of data leakage while maintaining scalability and adaptability.

At its core, FL operates by coordinating model training across distributed clients (e.g., devices, hospitals, or edge data centers) under the orchestration of a central parameter server. Variants of fully decentralized frameworks have also been proposed to eliminate single points of failure [176]. Depending on how features and samples are distributed, two main paradigms are distinguished: Horizontal Federated Learning (HFL) [177], where clients share the same feature space but hold different samples, and Vertical Federated Learning (VFL) [178], where clients share the same samples but different features.

Despite its advantages, FL faces two persistent challenges: *(i)* reduced model performance under statistically heterogeneous (i.e., non-Independent and Identically Distributed (non-IID)) data distributions, and *(ii)* limited interpretability in most state-of-the-art neural network-based approaches. These issues are particularly critical in regulated or safety-critical domains such as healthcare, finance, and IoT-based monitoring, where transparent decision-making is essential. In such contexts, black-box federated models like Federated Averaging (FedAvg) and Federated Proximal optimization (FedProx) are difficult to justify or audit, especially under privacy regulations that emphasize accountability [179, 180].

To address these limitations, this chapter explores a rule-based FL framework designed to achieve interpretable model behavior and stable training under heterogeneous conditions, while preserving data privacy, ownership, and security.

7.1.2 Contributions

Building on recent advances in interpretable federated systems [181], this chapter investigates the potential of Classification And Regression Trees (CART)-based decision tree models for FL. Unlike DNN-based approaches, the proposed framework emphasizes lightweight, transparent, and rule-based learning mechanisms that are particularly suited for environments with heterogeneous data distributions. The main contributions are summarized as follows:

1. Introduction of Federated Learning for Interpretable Rule Trees (FL-IRT), a novel FL framework based on Decision Tree (DT) models (using the CART algorithm), specifically designed to handle non-IID data while ensuring transparency and interpretability through rule-based decision logic.

2. Development of a rule-sharing mechanism in which human-readable decision rules are extracted from client-level DTs and aggregated at the server by labeling a synthetic dataset. This process yields a globally consistent and interpretable model.

3. Benchmarking against traditional federated baselines (e.g., FedAvg, FedProx) and hybrid rule-based frameworks such as Global-Local Rule-Based eXplanations for Federated Learning (GLOR-FLEX), highlighting performance, interpretability, and computational efficiency.

4. Demonstration that FL-IRT achieves superior performance in terms of computational speed and Accuracy (ACC) under heterogeneous conditions, while uniquely offering explainability through traceable rule sets. In addition, the framework outperforms hybrid models such as GLOR-FLEX, which still depend on rule extraction from neural networks.

5. Provision of both quantitative and qualitative evidence of how the framework adapts to heterogeneous clients, enabling deeper insight into the global decision-making process in federated settings.

7.2 Methodology

The FL-IRT algorithm enables interpretable global models by combining local rule extraction with server-based synthetic data generation and secure aggregation. The framework, sketched in Fig. 7.1, includes a central server (i.e., the Parameter Server (PS)) and Q cooperating clients. Without exchanging client data that remain local to them, this architecture is designed to ensure privacy preservation, model transparency, and robustness under data heterogeneity without requiring surrogate training or complex post-hoc rule merging. The PS handles communication, manages the global rules and their distribution to the clients, and reorganizes local rules while preserving data privacy during each client's local learning process.

In this study, DTs models with the CART algorithm are adopted as the foundational paradigm for both client- and server-side learning, leveraging their interpretability and compatibility with rule extraction. DTs trained with CART are widely recognized supervised ML classification methods [182], designed to partition data into more homogeneous

The content of this chapter has been published in the following paper:

A. Samandari, M. Marchese, A. Paglialonga, F. Patrone, V. Rampa, and M. Mongelli, "Federated Learning for Rule-Based Systems: Preliminary Studies," 2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI), pp. 143–148, Sep. 2024, IEEE. doi: 10.1109/RTSI60181.2024.10704658.

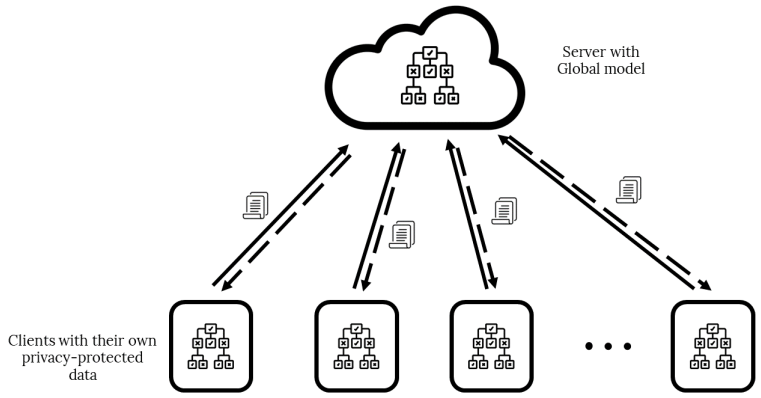


Figure 7.1: Overview of the FL-IRT framework.

groups. Moreover, DT-based models [183] require less computational power and a limited memory footprint compared to other ML methods.

A general DT model, with input features $x_i \in X$ and labels $y_i \in Y$, generates a function $f : \mathbb{R}^d \rightarrow \{0, \ldots, q-1\}$, where d is the number of attributes (i.e., features) and q the number of labels, while X and Y are the sets of the input features and output labels. In this study, both local and global DT models were employed, focusing on binary and multi-class classification scenarios. As detailed in Sect. 7.3.1, four different datasets were used for learning, verification, and testing, with their characteristics summarized in Tab. 7.1. The CART algorithm [184], a popular algorithm for constructing DTs, was exploited for the learning steps as well.

The FL-IRT framework integrates two complementary privacy-preserving mechanisms to ensure secure communication and protect local data distributions. To this aim, Transport Layer Security (TLS) is applied to client–server exchanges, encrypting rules and metadata to prevent interception or tampering. In addition, Differential Privacy (DP) is enforced at the client side before transmitting rules and metadata to the server and vice versa. This obfuscation prevents potential leakage of individual-level information from rule statistics while preserving the overall utility of the aggregated model [174, 185].

7.2.1 Definitions

Some definitions are included below to better detail the proposed algorithms.

Definition 7.2.1 (Data range conditions)**.** These conditions refer to the boundary constraints, represented as $x_i \leq Max_i$ and $x_i \geq Min_i$ inequalities, that are associated with the

input numerical features. In more details, for a given feature x_i, range conditions take the form of rules that enforce the given constraints $x_i \leq Max_i$ AND $x_i \geq Min_i$ or, equivalently, $Min_i \leq x_i \leq Max_i$. These constraints are enforced during local DT construction to reflect client-specific data bounds. It is worth noticing that these conditions are selectively added to the rule set only if they are not already present in the extracted rules. Usage of range conditions ensures that the resulting DT models accurately include the range of feature values within each client unique distribution. According to this, range conditions for the feature x_i can be expressed as:

$$r_k \wedge (x_i \leq Max_i) \wedge (x_i \geq Min_i) \tag{7.1}$$

This verbose rule extension guarantees that local models observe the specific lower (Min_i) and upper (Max_i) bounds of each (possibly non-IID) i-th feature across clients. Range conditions are not typically useful at the local level for improving model ACC (they do not improve model accuracy), but they are sent to the PS to build a global knowledge about the behavior of the nodes. This is the most critical step to recover the variables inherent to non-IID conditions. For categorical features, the client builds the histogram Hist of the category counts. More details will be given in the next section.

Definition 7.2.2 (Rule relevance)**.** It is a key concept in rule-based learning, where, to each rule, it is assigned a value in the [0, 1] range that corresponds to its predictive ability [186]. It is computed by combining the covering $C(r_k)$ and the error $E(r_k)$ metrics, commonly known as the True Positive Rate and False Positive Rate of the rule r_k, respectively. Specifically, the rule relevance R_k of the rule r_k is defined as:

$$R_k = R(r_k) = C(r_k) \cdot (1 - E(r_k)) \tag{7.2}$$

with:

$$C(r_k) = \frac{TP(r_k)}{TP(r_k) + FN(r_k)}$$

$$E(r_k) = \frac{FP(r_k)}{FP(r_k) + TN(r_k)}$$

Denoting $\hat{y}_i$ as the class label predicted by the rule r_k for the couple (x_i, y_i), $TP(r_k)$ and $FP(r_k)$ are defined as the number of instances that correctly and wrongly satisfy rule r_k, i.e. if $\hat{y}_i = y_i$ and $\hat{y}_i \neq y_i$, respectively. Likewise, $TN(r_k)$ and $FN(r_k)$ represent the

number of samples (x_i, y_i) which do not meet at least one condition in rule r_k, with $\hat{y}_i \neq y_i$ and $\hat{y}_i = y_i$, respectively.

7.2.2 Client-side Algorithm

As outlined in Algorithm 1, after initialization, for each n-th communication round, each client node computes auxiliary data range information. In fact, to support downstream rule evaluation and synthetic data generation, each client also extracts local feature statistics. For each numerical feature x_i, the client computes the local minimum and maximum values over the n-th batch set extracted from X. For each categorical feature, the client builds a histogram of category counts using the function **BuildHistogram**(). The resulting structures $Min(n)$, $Max(n)$, and $Hist(n)$ summarize local distributions and are sent to the server to assist global learning and sampling.

Then the local DT model is trained by extracting batch of data samples from the local dataset. The subsequent phase involves the extraction of the base rule set $\{r_k\}$ from the DT by leveraging their tree-shaped structure. The extraction of the rule set from a DT implies the interpretation of the tree branching structure. At each internal node, a decision is made based on a specific attribute, thus leading to branches corresponding to conditions about the attribute values. The tree leaves represent the classification outcomes. Rules are then extrapolated by traversing the tree from the root node to a leaf one by capturing the attribute conditions along the path. These conditions collectively define a rule. In the proposed framework, this rule extraction process is indispensable to translate DTs models into interpretable and actionable rule-based models according to the eXplainable Artificial Intelligence (XAI) paradigm. Not only the rule set is extracted by traversing the local DT model, but also the corresponding Rule Relevance Set $\{R(r_k)\}$ is evaluated by the **ComputeRuleRel**() function. It is worth noticing that each client sets up a threshold η for rule relevance checking ($\eta = 0.2$ in the following sections). Rules with a rule relevance greather or equal to η are added to the Rule Relevance Set by the **AddRuleRel**() procedure.

Then, **DPSanitize**() applies Differential Privacy (DP) to local statistics data and Rule Relevance values before transmission. It perturbs each input values var generating the output value var_{dp} by adding to the input value a certain amount of zero mean Laplace noise with the scale parameter set according to $1/\epsilon$ ($\epsilon = 0.5$ in the following sections):

$$var_{dp} = \textbf{DPSanitize}(var) = var + \text{Laplace}(0, 1/\epsilon) \tag{7.3}$$

Algorithm 1: Client Algorithm

N: Number of rounds
$X(n)$: Dataset of features samples available in the batch of round n
$Y(n)$: Label samples associated to $X(n)$ (batch of round n)
RuleSet: Rule Set
RuleRelSet: Rule relevance Set
DT: Decision Tree model
R_k: k-th Rule Relevance
r_k: k-th Rule
LocalModel: Local DT model
GlobalModel: Global DT model
$\eta = 0.2$: Rule relevance threshold
$\epsilon = 0.5$: DP parameter

Initialize LocalModel, RuleSet, RuleRelSet, $Min(n), Max(n), Hist(n) \leftarrow \emptyset$
for $n = 1, \ldots, N$ **do**
  for each x_i, y_i **in** $(X(n), Y(n))$ **do**
    if x_i is numerical **then**
      $Min(n) \leftarrow \min(x_i)$
      $Max(n) \leftarrow \max(x_i)$
    else if x_i is categorical **then**
      $Hist(n) \leftarrow$ **BuildHistogram**(x_i)
    end if
    $(x_{\text{train}}, y_{\text{train}}) \leftarrow$ **SelectSamples**(x_i, y_i)
    LocalModel $\leftarrow$ **TrainDT** $(x_{\text{train}}, y_{\text{train}})$
    RuleSet$(n) \leftarrow$ **ExtractRulesFromModel**(LocalModel)
    for each r_k **in** RuleSet(n) **do**
      $R_k \leftarrow$**ComputeRuleRel**(r_k)
      if $R_k \geq \eta$ **then**
        RuleRelSet$(n) \leftarrow$ **AddRuleRel**$(R_k,$ RuleRelSet$(n))$
      end if
    end do
    $Min_{dp}(n) \leftarrow$ **DPSanitize**$(Min(n),\epsilon)$
    $Max_{dp}(n) \leftarrow$ **DPSanitize**$(Max(n), \epsilon)$
    $Hist_{dp}(n) \leftarrow$ **DPSanitize**$(Hist(n), \epsilon)$
    $R_{dp}(n) \leftarrow$ **DPSanitize**(RuleRelSet$(n), \epsilon)$
    SendToServerUsingTLS$(Min_{dp}(n), Max_{dp}(n), Hist_{dp}(n),$ RuleSet$(n), R_{dp}(n))$
    ReceiveFromServerUsingTLS(GlobalModel)
    LocalModel $\leftarrow$ GlobalModel
    end do
end do

This ensures that sensitive client information remains protected while preserving the overall utility of the transmitted data for global aggregation.

Then, DP-protected data are sent yo the PS by encrypting them with the Transport Layer Security (TLS) protocol using the **SendToServerUsingTLS**() function*. By exploiting the same encryption protocol, each client receives the global model from the PS using the **ReceiveFromServerUsingTLS**() function†. These details are outside the scope of this study. Finally, the global model overwrite the old local model and another turn begins. The number of rounds is set to N.

7.2.3 Server-side Algorithm

The pseudo-code of Algorithm 2 sketches the FL-IRT method implemented by the PS. In the proposed framework, the central server PS constructs an interpretable global model by labeling synthetic data using aggregated rules from the selected local clients.

To this aim, during each communication round, the PS randomly selects, using the **RandomlySelectClientSubsetl**() function, a subset of M out of Q clients, with each client having an equal probability of selection. The PS then receives from each m-th client, using the **ReceiveFromClientUsingTLS**() function‡, the *range rules* corresponding to the Min_i and Max_i values of each x_i feature (arranged in two vectors Min and Max having length equal to d). The *global* range rules $GbMin_i$ and $GbMax_i$ for each x_i feature are then obtained as $GbMin_i = \min(\{Min_i(m)\}_{m=1}^{M})$ and $GbMax_i = \max(\{Max_i(m)\}_{m=1}^{M})$.

In parallel, it receives also the rule relevance from each selected client, and then it aggregates them to form the *global* rule set $GbRuleSet$. Subsequently, the PS generates some random feature samples based on the uniform distribution $\mathcal{U}(\mathrm{GbMin}_i, \mathrm{GbMax}_i)$ for each i-th feature. However, if known, an appropriate feature distribution $\mathcal{D}_i$ is assumed instead of the uniform one.

In this study, instead of relying solely on uniform random sampling, FL-IRT introduces a confidence-aware sampling strategy guided by the rules received from clients. For each feature, the server extracts threshold values from high-relevance rules and uses them to bias the generation of synthetic points. Specifically, the server samples a portion of synthetic data around these informative thresholds using Gaussian distributions centered at each rule condition.

*The client-server handshaking is simplified here since only M out of Q clients are randomly selected, then interrogated to send DP-protected and other data to the server. In addition, client-server synchronization methods depends on the specific hardware/software architecture implementation.

†A proper handshaking protocol guarantees client-server synchronization for each round n

‡See also the previous two footnotes about the simplified handshaking between clients and server.

Algorithm 2: Server Algorithm

L: Number of class labels
N: Number of rounds
Q: Number of clients
M: Number of randomly selected clients for learning
N_{SR}: Number of generated samples per round
$GbRuleSet$: Global Rule Set
$GbRuleRelSet$: Global Rule Relevance Set
$GbSamples$: Global Synthetic Dataset
$GbHistSet$: Global histogram set
$GbMin$: Global minimum set
$GbMax$: Global max set
$BoostCentersSet$: means of density function for sampling
$PMFSet$: Probability Mass Function
$Samples$: Server-generated synthetic samples
r_k: k-th Rule

Initialize $GbRuleSet, GbRuleRelSet, GbPoints, GbHistSet, GbMin, GbMax \leftarrow \emptyset$
for $n = 1, \ldots, N$ **do**
  $ClSubSet(n) \leftarrow$ **RandomlySelectClientSubset**(M, Q)
  for each $m \in ClSubSet(n)$ **do**
    $Min_{dp}(m), Max_{dp}(m), Hist_{dp}(m)$, RuleSet$(m)$, $R_{dp}(m) \leftarrow$ **ReceiveFromClientUsingTLS**(m)
  end do
  $GbRuleSet \leftarrow \bigcup_{m=1}^{M}$ RuleSet(m)
  $GbRuleRelSet \leftarrow \bigcup_{m=1}^{M} R_{dp}(m)$
  $GbHistSet \leftarrow \bigcup_{m=1}^{M} Hist_{dp}(m)$
  $GbMin \leftarrow \min(\{Min_{dp}(m)\})_{m=1}^{M}$
  $GbMax \leftarrow \max(\{Max_{dp}(m)\})_{m=1}^{M}$
  $BoostCentersSet \leftarrow$ **ExtractThresholdsFromRules**$(GbRuleSet, GbRuleRelSet)$
  $PMFSet \leftarrow$ **BuildGlobalPMF**$(GbHistSet)$
  $Samples \leftarrow$ **ConfidenceAwareSampling**$(GbMin, GbMax, BoostCentersSet, PMFSet, N_{SR})$
  for $i = 1, \ldots, N_{SR}$ **do**
    Initialize $Matched \leftarrow$ false
    for $l = 1, \ldots, L$ **do** $WeightedVote[l] = 0$
    for each $r_k \in GbRuleSet$ **do**
      if SampleSatisfyRule$(r_k, Samples(i))$ **then**
        $\hat{y}_k \leftarrow$ **PredictLabel**$(r_k, Samples(i))$
        $WeightedVote[\hat{y}_k] += R_{dp}(r_k)$
        $Matched \leftarrow$ true
      end if
    end do
    if $Matched$ **then**
      $\hat{y}_i \leftarrow \arg\max_l (WeightedVote)$
    else
      $\hat{y}_i \leftarrow$ **FallbackClassFromGlobalStats**
    end if
  end do
  $GbSamples \leftarrow \bigcup_{i=1}^{N_{SR}} (Samples(i), \hat{y}_i)$
  GlobalModel $\leftarrow$ **TrainDT**$(GbPoints)$
  SendToClientsUsingTLS(GlobalModel)
end do

This approach increases the density of synthetic points in semantically important regions of the feature space, improving the coverage of meaningful decision boundaries. This strategy, implemented by the **ExtractThresholdsFromRules**() and **ConfidenceAwareSampling**() procedures, ensures better alignment between the synthetic dataset and the logic expressed in the aggregated rule set $GbRuleSet$, leading to more accurate and stable global models.

These two procedures, designed during the preliminary tests to improve the system performances, are detailed below:

> $BoostCentersSet$ contains the thresholds of high-relevance rules received from clients by the **ExtractThresholdsFromRules**() function. It is a server-side procedure where the term $Boost$, defined as in [187], denotes that the extracted thresholds drive the generation of synthetic data by increasing Gaussian sample density near those thresholds, being each threshold a decision boundary for class assignment.
>
> **ConfidenceAwareSampling**() is a server-side function for synthetic data generation that biases sample generation toward informative regions in the feature space. For numerical features, 30% of the samples are drawn from Gaussian distributions centered on $BoostCenters$, while the remaining 70% are drawn uniformly from the global range [GbMin, GbMax]. For categorical features, values are sampled from global Probability Mass Functions (PMFs) [188], that are constructed by aggregating the local histograms $Hist(m)$ received from all selected clients.

The global rule set is then employed to label each random sample using a function called **PredictLabel**(), which acts as a logic machine predicting labels that satisfy the rules. The server then aggregates the predicted labels using a weighted majority voting mechanism based on the relevance of matching rules, yielding the final label $\hat{Y}$ for each synthetic sample.

To assign a label to each synthetic sample, the server aggregates predictions from all matching rules using a weighted voting mechanism. Each rule r_k contributes to the vote for its predicted label $\hat{y}_k$ proportionally to its differentially private relevance score $R_{dp}(r_k)$. The label $\hat{Y}$ with the highest accumulated vote is selected. This process favors high-quality rules while mitigating the impact of noisy or weak ones. If no rule in the global rule set matches a given synthetic sample, the server uses a fallback mechanism implemented via **FallbackClassFromGlobalStats**(). This function selects a label based on global statistics, such as majority class distribution among confident predictions. It guarantees full label coverage and enables generalization to out-of-distribution regions.

The global synthetic dataset $GbPoints$ consists of labeled samples generated by sampling the server-generated $Samples$ from the global feature distributions (numerical and categorical) and labeling them via weighted rule voting. Each i-th entry in $GbPoints$ is a tuple $(Samples(i),\ \hat{y}_i)$. The dataset is constructed to ensure semantic consistency—neighboring samples reflect the same rule-based logic unless separated by clear decision boundaries.

Finally, the global model is a DT trained by the server using the synthetic dataset $GbPoints$. It synthesizes distributed decision logic across clients into a centralized, interpretable structure. Once trained, the global model is distributed back to all clients, completing one round of the FL-IRT communication protocol.

The proposed FL-IRT method is validated across four datasets in Sect. 7.3.6, where its performance is compared against FedAvg, FedProx, Hierarchical crOss-siLo feDerated Averaging (HOLDA), Global-Explainer (G-Explainer), and Server-TREPAN (S-TREPAN). The last three models are derived from the GLOR-FLEX framework.

7.3 Experimental validation

This section describes the experimental validation of the proposed FL-IRT framework. The evaluation is carried out by comparing FL-IRT against established FL baselines and explainability-focused alternatives. The presentation begins with the datasets, hardware and software configurations, and implementation details. Subsequently, the comparative models are introduced, namely FedAvg, FedProx, and three variants derived from the GLOR-FLEX framework: HOLDA, G-Explainer, and S-TREPAN. Finally, a detailed analysis is reported, covering ACC, F1-score, Matthews Correlation Coefficient (MCC), computational efficiency, explainability, privacy-preserving mechanisms, convergence behavior, and architectural complexity. This comprehensive evaluation highlights the strengths and limitations of FL-IRT under realistic FL scenarios.

7.3.1 Dataset

A series of simulations was conducted in different scenarios (i.e., use cases), employing four distinct datasets namely the Hospital dataset (HSP), Vehicle Platooning dataset (VP), Modified National Institute of Standards and Technology dataset (MNIST), and Adult dataset (ADL) datasets. The specific characteristics of these datasets are summarized in Tab. 7.1 and detailed below.

Table 7.1: Dataset Information.

Dataset	Clients (Q)	Samples	Samples/Client (B)	Features (d)	Categories (q)
HSP	190	38000	200	3	2
VP	108	21600	200	7	2
MNIST	86	25800	300	3	4
ADL	4	45220	11305	14	2

7.3.1.1 Hospital dataset (HSP)

The Hospital dataset, referred to as the HSP dataset, is a simple, synthetic but extremely non-IID dataset. The aim of using this dataset is to help the reader understand how the proposed method overcomes non-IID issues by inferring a proper rule set at the server side. The dataset contains patient records from two distinct hospitals. In this dataset, that was ad-hoc generated for this study, each patient has $d = 3$ features: X_1, X_2, and *Age*. Patients are distributed across two hospitals, each with similar features (X_1, X_2), ensuring Identically Distributed (IID) characteristics within each hospital. However, the third feature *Age* varies between the two hospitals, with patients in the first hospital (i.e., Group 1) falling between the ages of 20 and 30, and those in the second hospital (i.e., Group 2) between 60 and 70.

The distribution of the HSP dataset is homogeneous across the first two features, highlighting uniformity, while showing strong non-IID characteristics in the third feature *Age*. The binary labels in this dataset represent two distinct health statuses: *healthy* (i.e., label 0) and *unhealthy* (i.e., label 1).

The interpretation of these labels is reversed between the two hospitals: what is considered *healthy* in one hospital is labeled as *unhealthy* in the other, and vice-versa. The 3D scatterplot of the dataset is shown in Fig. 7.2, providing visual insight into the distribution patterns across the feature space.

5% of the entire dataset (consisting of 38000 distinct samples) was reserved for testing for testing, while the remaining 95% was distributed for training among clients.

7.3.1.2 Vehicle Platooning dataset (VP)

The Vehicle Platooning dataset, referred to as the VP dataset, was introduced by [189] as a representative benchmark for data-driven enhancements in autonomous driving systems [190]. The data-driven approach is motivated by the need to achieve an optimal trade-off between performance metrics, i.e. maximizing speed and minimizing the re-

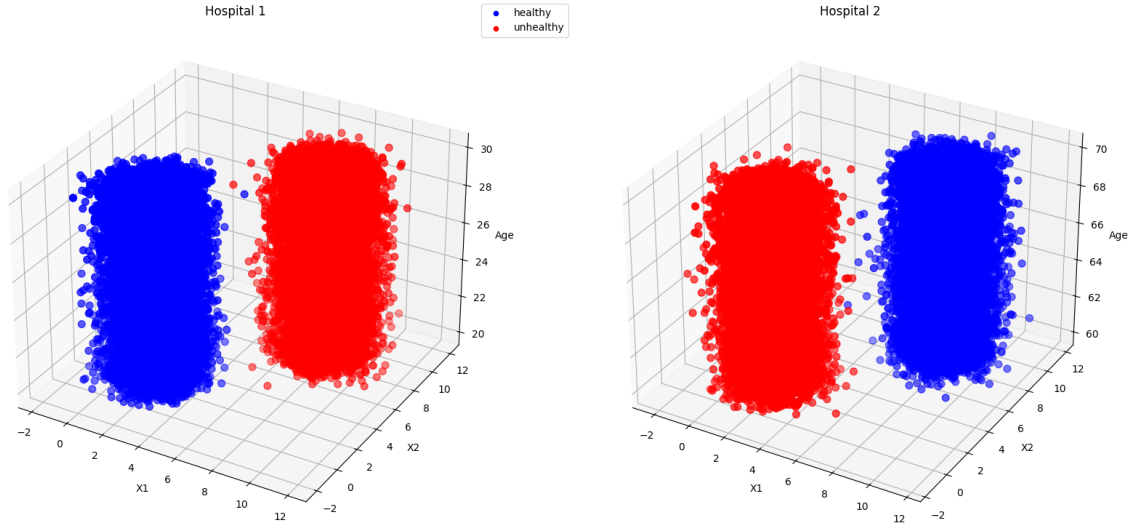


Figure 7.2: 3D Scatterplot of the HSP dataset: homogeneous distribution in the two features X_1, X_2 with non-IID patterns in the *Age* feature. The PS has to learn the clusters shift, depending on *Age*.

ciprocal distance between vehicles, and safety considerations, i.e. collision avoidance, where the collision label 0 corresponds to no collision and the collision label 1 stands for collision. There are no analytical solutions to this problem. The dataset is mostly IID across seven features, with six of them (d_0, F_0, N, d_{ms}, PER, and v_0) showing consistent distributions. However, the feature m exhibits non-IID characteristics due to a systematic distributional shift across client groups.

The non-IID nature of the dataset is illustrated in Fig. 7.3, which shows the attribute relevance rankings for the collision=1 class. The differences between Group 1 and Group 2 clearly demonstrate that each group learns distinct rule sets due to divergent statistical behaviors. The dataset is distributed among the two groups of clients, each consisting of 54 clients, according to the m values: Group 1 holds data for $m \leq 1000$, whereas Group 2 holds data for $m > 1000$. A total of 5% of the dataset was reserved for testing, while the remaining 95% was distributed among the training clients.

7.3.1.3 MNIST dataset (MNIST)

The MNIST [191] dataset was considered as the third non-IID scenario. The UMAP dimensionality reduction technique [192, 193] was applied to project the original high-dimensional MNIST images into a four-dimensional numerical feature space. For Group 1, the target classes 3 and 5 were selected, corresponding to the images of digits "3" and

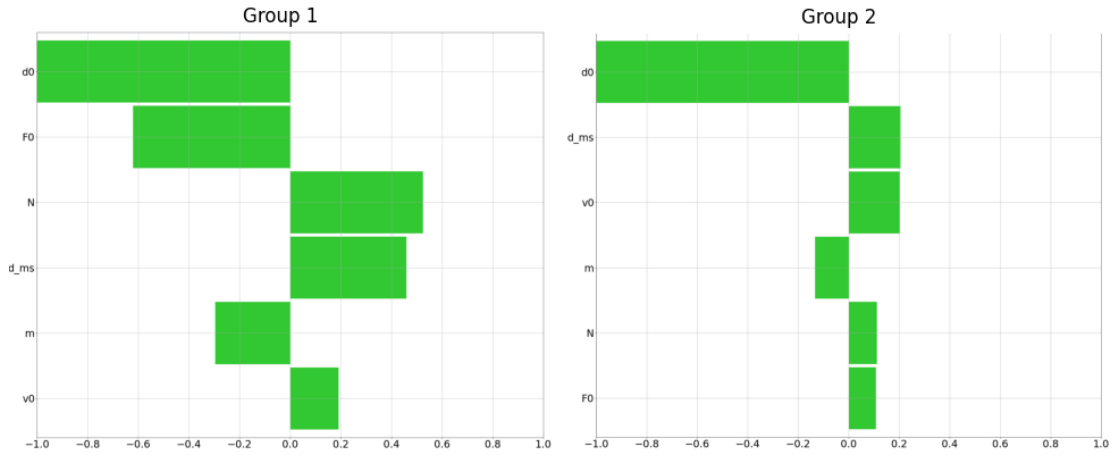


Figure 7.3: Examples of attribute relevance rankings for collision (i.e., collision label equal to 1) are depicted for Group 1 (left) and Group 2 (right), concerning the *Vehicle Platooning* dataset. Different rankings mean different rule sets, typically motivated by different statistical behaviors of data (between the two groups).

"5". Conversely, Group 2 included the target classes 0 and 7, corresponding to the digits "0" and "7". Consequently, each client addressed a binary classification task, while the PS was required to solve a four-class classification problem.

7.3.1.4 Adult dataset (ADL)

The fourth and final dataset used in the simulations is the ADL dataset [194], a widely adopted benchmark in machine learning for income classification. This dataset consists of census records comprising both numerical and categorical features. After pre-processing, which involved removing incomplete entries and encoding categorical attributes into numerical format, a total of 45,220 samples were retained. Each record includes 14 attributes related to demographics and employment, and the binary label indicates whether a person's income exceeds 50k$ per year. All categorical variables were encoded numerically using label encoding. An IID setup was adopted for this dataset, in which the preprocessed records were randomly and equally distributed across clients, ensuring no overlapping samples between different clients.

7.3.2 Dataset Statistical Properties

Across the four datasets, a wide range of statistical properties was intentionally covered to comprehensively evaluate the robustness of the proposed FL-IRT framework in comparison with other federated learning methods. Specifically, the HSP, VP, and MNIST datasets exhibit strong non-IID characteristics, with data heterogeneity stemming from feature shifts, group-wise label inconsistencies, and class disjointness. In contrast, the ADL dataset presents a highly imbalanced yet IID scenario, posing challenges in learning from skewed label distributions. Additionally, the MNIST dataset introduces an image-derived feature space via UMAP transformation, allowing evaluation of FL-IRT on visual data represented in a low-dimensional numerical form. These diverse evaluation scenarios ensure that FL-IRT is tested under realistic and challenging federated learning conditions.

7.3.3 Baseline and Comparative Models

The performance of the proposed FL-IRT framework was compared against several established FL models. Specifically, the following baseline and comparative approaches were considered:

Federated Averaging (FedAvg). FedAvg [32] is the foundational algorithm in FL and serves as a widely adopted baseline in the literature. In FedAvg, each client trains a local model on its private dataset and transmits the resulting model parameters to the central server. The server then aggregates these parameters by computing their average, forming a new global model. This process is repeated over multiple communication rounds until convergence.

Federated Proximal (FedProx). FedProx [34] extends FedAvg by introducing a proximal term in the clients' local objective function. This proximal term penalizes large deviations of the local models from the current global model, thereby reducing client drift and improving stability in heterogeneous, non-IID environments.

GLOR-FLEX (Global-Local Rule-Based eXplanations for Federated Learning). GLOR-FLEX [127] is an explainable FL framework that shares conceptual similarities with FL-IRT, particularly in its focus on rule-based modeling and privacy-preserving explainability. The framework includes three core components:

HOLDA is a DNN model trained in a cross-silo federated setting, where the global model is updated by selecting the locally best-performing model (based on validation accuracy) rather than applying parameter averaging, as in FedAvg. This approach improves stability in heterogeneous environments.

G-Explainer (TREPAN + HOLDA + GLocalX methods) is a global rule-based explainer constructed by aggregating symbolic rules extracted via TREPAN from each client's local HOLDA model. These rules are sent to the server and merged using GLocalX (GLObal to loCAL eXplainer), which filters and prioritizes them based on fidelity and coverage.

S-TREPAN (HOLDA + TREPAN methods) is a surrogate DT trained centrally on synthetic data sampled from client-side Gaussian Mixture Models (GMMs). The clients fit GMMs locally and send the resulting synthetic records to the server, which labels them using the global HOLDA model. The labeled dataset is then used to train a symbolic surrogate tree, enabling global interpretability.

While the original GLOR-FLEX work primarily focused on evaluating HOLDA, the comparison in this study was extended by also assessing G-Explainer and S-TREPAN on real test data. These rule-based models serve as strong interpretability-focused methods that align closely with the goals of FL-IRT.

7.3.4 Hardware and Software Setup

All FL algorithms used in the simulations were implemented and executed in a local computing environment equipped with an Intel(R) Core(TM) i7-13700H processor running at 2.4 GHz (14 physical and 20 logical cores), 16 GB of RAM, and a 2 TB SSD. The system included both an integrated Intel(R) Iris(R) Xe Graphics GPU and a dedicated NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB of VRAM.

All simulations were conducted using Python v3.10.14, with the following major libraries: TensorFlow v2.10.1, SciPy v1.13.1, scikit-learn v1.5.0, pandas v2.2.2, and numpy v1.26.4. The FL-IRT framework and all comparative models were primarily executed on CPU, except for the DNN components that opportunistically utilized GPU acceleration where supported by the underlying libraries.

7.3.5 Simulation Settings and Model Configurations

All FL simulations were conducted using a single PS and a number M of clients randomly selected from the total pool of Q clients, depending on the dataset. For the HSP, VP, and MNIST datasets, three different configurations were evaluated: 6, 12, and 24 clients per round. In these cases, clients were sampled evenly from two disjoint groups ($M/2$ clients

per group). For the ADL dataset, which includes only $Q = 4$ clients, all clients were used in every communication round (i.e., $M = Q$).

Hospital Dataset. For the HSP dataset ($Q = 190$), the FedAvg and FedProx models employed a DNN architecture consisting of three hidden layers with 25, 60, and 25 neurons, respectively. Stochastic Gradient Descent (SGD) was adopted as the local optimizer with a learning rate of 0.01. The batch size B per client per round was fixed at 200.

Vehicle Platooning Dataset. For the VP dataset ($Q = 108$), a fully connected DNN with three hidden layers containing 35, 100, and 35 neurons, respectively, was used. The Adam optimizer was applied for both FedAvg and FedProx, with a learning rate of 0.01. The FedProx hyperparameter was fixed at $\mu = 0.01$. The batch size per client was set to 200.

MNIST Dataset. For the MNIST dataset ($Q = 86$), a DNN with three hidden layers of 30, 60, and 30 neurons was used, followed by a softmax output layer with 4 neurons (for multi-class classification). Stochastic Gradient Descent (SGD) was adopted as the optimizer with a learning rate of 0.01, and the FedProx hyperparameter was set to $\mu = 0.001$. The batch size was set to 34 per client per round.

Adult Dataset. For the ADL dataset ($Q = 4$), a fully connected DNN architecture with hidden layers of 35, 100, and 35 neurons, respectively, was employed. SGD was adopted as the optimizer with a learning rate of 0.01, and the FedProx hyperparameter was set to $\mu = 0.001$. The batch size per client was equal to 11,305 (i.e., one full local dataset per round).

HOLDA and GLOR-FLEX Models. The HOLDA model used across all datasets except MNIST consisted of a fully connected DNN with two hidden layers of 256 neurons each, followed by a softmax output layer (2 neurons for binary classification). This method was not applied to MNIST, as it is a multi-class dataset. To ensure consistency with the GLOR-FLEX setup [127], the same training strategy was adopted: 10 global communication rounds, each comprising 3 local training epochs, followed by 10 client-side fine-tuning epochs. G-Explainer and S-TREPAN were constructed using synthetic datasets of 1000 samples per client, obtained by sampling from Gaussian Mixture Models (GMMs) with 5

components fitted on local data. These configurations, originally introduced for the ADL dataset, were extended to all datasets for consistent and fair benchmarking.

The HSP and VP datasets used in these experiments are available at [195].

Communication Round Semantics. In the FL-IRT, FedAvg, and FedProx frameworks, a single communication round is defined as the cycle in which selected clients train local models, send updates to the server, receive the updated global model, and integrate it into the next training phase. This definition ensures consistent timing and synchronization across rounds. By contrast, GLOR-FLEX follows a different timing structure: the global DNN is updated once every 3 local training epochs per round and is followed by 10 additional local fine-tuning epochs at each client. The symbolic explanation models (G-Explainer and S-TREPAN) are constructed after model convergence in a decoupled post-hoc stage, using synthetic GMM-based samples.

7.3.6 Results

The performances of FL-IRT, FedAvg, FedProx, and three models derived from the GLOR-FLEX framework i.e. HOLDA (the neural model trained via model selection), G-Explainer (symbolic rule merging using HOLDA and GLocalX), and S-TREPAN (a surrogate DT trained using HOLDA and GMM-based synthetic data), were evaluated using multiple classification metrics: ACC, F1-score, and MCC, computed on the four test datasets. A detailed discussion about the performances of the compared frameworks is given in Sects. 7.3.6.1–7.3.6.4.

ACC quantifies the overall proportion of correct predictions and it is defined as:

$$\text{ACC} = \frac{TP + TN}{FN + FP + TN + TP} \tag{7.4}$$

F1-score, the harmonic mean of precision and recall, captures the balance between false positives and false negatives:

$$\text{F1-score} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{7.5}$$

MCC provides a more balanced evaluation in the presence of class imbalance by considering all four confusion matrix components:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7.6}$$

Together, these metrics provide a comprehensive assessment of predictive accuracy, balance, and robustness across all datasets. Both quantitative and qualitative results are reported below, including comparisons of runtime performance.

Unless otherwise specified, all computational times reported in this section for the FL-IRT framework refer to simulations that do not include any privacy-preserving mechanisms such as TLS or DP. This is done for fair comparison against the other methods that do not include any privacy-preserving procedures. However, the overhead introduced by TLS and DP is explicitly analyzed in Subsection 7.3.6.6.

7.3.6.1 Hospital Dataset

The FL-IRT framework demonstrates near-perfect performance on the HSP across all evaluated client configurations (6, 12, and 24 clients). As shown in Tab. 7.2, FL-IRT consistently outperforms FedAvg, FedProx, and all GLOR-FLEX variants (HOLDA, G-Explainer, and S-TREPAN) in ACC, F1-score, and MCC metrics, with minimal standard deviation across rounds. Specifically, FL-IRT achieves accuracies about 0.998% with corresponding MCC values above 0.996, even as the number of participating clients increases.

Table 7.2: HSP dataset: evaluation metrics averaged over the last 10 communication rounds. Values are reported as mean ± standard deviation.

Method	**6 Clients**			**12 Clients**			**24 Clients**		
	Acc	**F1-Score**	**MCC**	**Acc**	**F1-Score**	**MCC**	**Acc**	**F1-Score**	**MCC**
FL-IRT	0.998 ± 0.003	0.998 ± 0.003	0.996 ± 0.006	0.999 ± 0.000	0.999 ± 0.000	0.998 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
FedAvg	0.954 ± 0.019	0.954 ± 0.019	0.910 ± 0.036	0.961 ± 0.018	0.961 ± 0.018	0.924 ± 0.034	0.962 ± 0.019	0.963 ± 0.019	0.928 ± 0.034
FedProx	0.958 ± 0.016	0.958 ± 0.016	0.919 ± 0.029	0.970 ± 0.012	0.970 ± 0.012	0.942 ± 0.023	0.978 ± 0.013	0.978 ± 0.013	0.958 ± 0.025
HOLDA	0.752 ± 0.000	0.735 ± 0.000	0.578 ± 0.000	0.838 ± 0.090	0.829 ± 0.098	0.717 ± 0.148	0.894 ± 0.073	0.891 ± 0.079	0.808 ± 0.118
G-Explainer	0.510 ± 0.002	0.510 ± 0.002	0.020 ± 0.003	0.511 ± 0.001	0.511 ± 0.001	0.021 ± 0.002	0.510 ± 0.001	0.510 ± 0.002	0.021 ± 0.004
S-TREPAN	0.749 ± 0.009	0.732 ± 0.009	0.569 ± 0.027	0.836 ± 0.094	0.828 ± 0.104	0.703 ± 0.160	0.923 ± 0.087	0.920 ± 0.093	0.862 ± 0.144

Fig. 7.4 presents the results for the 24-client configuration across all communication rounds. FL-IRT maintains a stable, high-performing curve across all metrics and converges early. On the contrary, FedAvg and FedProx exhibit slower convergence and greater fluctuations. HOLDA and S-TREPAN show improved performance with more clients but remain notably below FL-IRT. G-Explainer performs poorly, consistently hovering near random-guessing levels. This under-performance behavior is largely due to the absence of *Age*-based conditions in the local rule sets, an important predictive feature, leading to a global rule set that lacks this discriminative signal. The robustness of FL-IRT is further confirmed by its stable MCC values, even under highly non-IID data distributions.

Notably, FL-IRT's ability to extract and integrate *Age*-dependent symbolic rules results in superior generalization and ACC capabilities compared to the other methods.

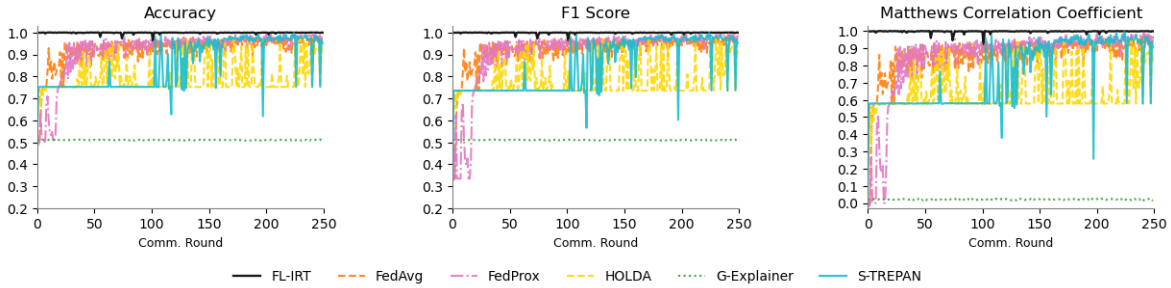


Figure 7.4: HSP dataset: test metrics across all communication rounds for the 24-client configuration.

Tab. 7.3 summarizes the runtime for each method. FL-IRT completes training up to 90% faster than FedAvg and more than 95% faster than FedProx. Despite running entirely on CPU, FL-IRT outperforms all GLOR-FLEX models—including HOLDA, G-Explainer, and S-TREPAN—which utilize GPU acceleration. These results highlight the computational efficiency of FL-IRT, even without specialized hardware.

Table 7.3: HSP dataset: processing time (in seconds) for different numbers of clients.

Method	**6 Clients (s)**	**12 Clients (s)**	**24 Clients (s)**
FL-IRT	179.3	190.0	204.1
FedAvg	2016.7	4022.9	7855.5
FedProx	6537.4	11328.8	21263.2
HOLDA	546.0	1074.2	2268.5
G-Explainer	421.2	828.7	1632.9
S-TREPAN	1107.0	2621.3	4530.5

7.3.6.2 Vehicle Platooning Dataset

The FL-IRT framework demonstrates consistently competitive performance on the VP dataset across all client configurations (6, 12, and 24 clients). As shown in Tab. 7.4, FL-IRT maintains the ACC above 0.847 and stable F1 and MCC scores, with low standard deviations across runs. It achieves early convergence with low variations, making it reliable even with a low number of participating clients.

Fig. 7.5 illustrates the 24-client scenario across the ACC, F1-score, and MCC metrics. The DNN-based methods achieve the best asymptotic results for ACC in the [0.864

Table 7.4: VP dataset: evaluation metrics averaged over the last 10 communication rounds. Values are reported as mean ± standard deviation.

Method	6 Clients			12 Clients			24 Clients		
	Acc	F1-Score	MCC	Acc	F1-Score	MCC	Acc	F1-Score	MCC
FL-IRT	0.847 ± 0.004	0.846 ± 0.004	0.692 ± 0.007	0.851 ± 0.007	0.851 ± 0.007	0.703 ± 0.014	0.856 ± 0.006	0.856 ± 0.006	0.713 ± 0.011
FedAvg	0.878 ± 0.012	0.877 ± 0.012	0.756 ± 0.024	0.864 ± 0.019	0.863 ± 0.020	0.733 ± 0.031	0.881 ± 0.007	0.881 ± 0.007	0.763 ± 0.013
FedProx	0.869 ± 0.025	0.867 ± 0.026	0.739 ± 0.045	0.858 ± 0.010	0.857 ± 0.010	0.719 ± 0.019	0.881 ± 0.006	0.880 ± 0.006	0.762 ± 0.011
HOLDA	0.840 ± 0.004	0.839 ± 0.004	0.680 ± 0.010	0.844 ± 0.013	0.843 ± 0.013	0.965 ± 0.016	0.846 ± 0.003	0.845 ± 0.003	0.695 ± 0.005
G-Explainer	0.717 ± 0.065	0.707 ± 0.066	0.449 ± 0.140	0.744 ± 0.059	0.738 ± 0.060	0.501 ± 0.121	0.760 ± 0.050	0.753 ± 0.053	0.535 ± 0.102
S-TREPAN	0.790 ± 0.032	0.785 ± 0.036	0.603 ± 0.039	0.791 ± 0.033	0.787 ± 0.038	0.603 ± 0.033	0.805 ± 0.007	0.803 ± 0.008	0.613 ± 0.012

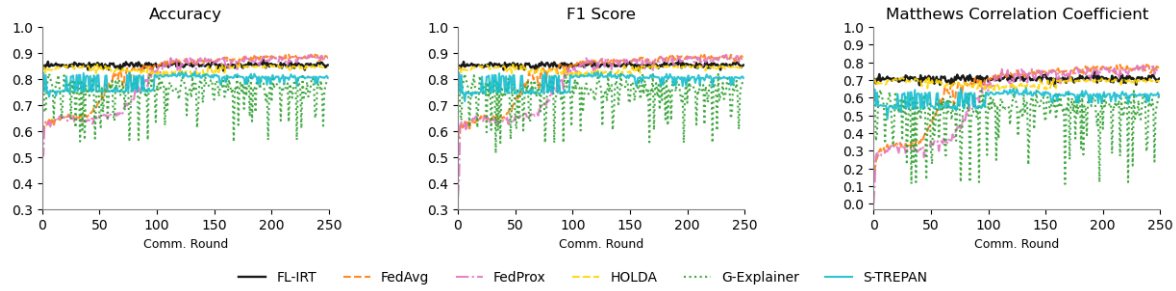


Figure 7.5: VP dataset: test metrics across all communication rounds for the 24-client configuration under statistical heterogeneity.

0.881] interval depending on the clients scenario. On the contrary FL-IRT achieves early convergence and then it maintains low variations across all metrics with values very close to the DNN-based ones. Among the GLOR-FLEX variants, HOLDA performs consistently and trails only slightly behind FL-IRT, while FedAvg reaches a marginally higher final ACC but with slower and less stable convergence. Moreover, S-TREPAN demonstrates more reliable behavior than G-Explainer, yet both remain less accurate and less stable than the DNN-based methods. It is worth noting that G-Explainer suffers from high volatility, especially in MCC, and struggles to generalize effectively across rounds.

Tab. 7.5 highlights the runtime efficiency of the proposed method. HOLDA achieves the best lower time for the 6 clients configuration (235.9 s) and the 12 clients one (419.0 s) but it lags behind FL-IRT for the 24 clients case. FL-IRT consistently exhibits a low processing time across all client configurations, completing the 24-client simulation in just 545.3 s that is the best time among all other methods for this scenario. In contrast, FedAvg and FedProx are approximately 70% and 95% slower, respectively, in the 6-client configuration. Notably, FL-IRT also outperforms the G-Explainer and S-TREPAN models, despite relying solely on CPU computation. In fact, all GLOR-FLEX models utilize GPU acceleration. These results confirm that FL-IRT's integrated and explanation-efficient design yields substantial computational savings, even without hardware-based optimization.

Table 7.5: VP dataset: processing time (in seconds) for different numbers of clients.

Method	**6 Clients (s)**	**12 Clients (s)**	**24 Clients (s)**
FL-IRT	485.5	507.5	545.3
FedAvg	1640.2	4424.2	6246.0
FedProx	8883.8	6684.9	16623.2
HOLDA	235.9	419.0	880.9
G-Explainer	728.0	1249.7	2530.6
S-TREPAN	1585.4	3160.0	6153.4

7.3.6.3 MNIST Dataset

The MNIST dataset poses unique challenges for FL methods due to its multi-class nature ($q = 4$) and statistically heterogeneous distribution across clients. Each group receives non-overlapping digits (e.g., digits "3" and "5" in Group 1 vs. "0" and "7" in Group 2), which amplifies non-IID effects and complicates classification—especially under limited client participation. It is worth noticing that since GLOR-FLEX-derived methods cannot handle multi-class datasets, they will not be included in the list of methods shown in Fig. 7.6 and Tabs. 7.6 and 7.7. This fact means that FL-IRT has an additional advantage over the considered rule-based methods for its ability to handle multi-class dataset as well.

As shown in Tab. 7.6, FL-IRT achieves competitive performance across all configurations, despite the problem complexity. While FedAvg and FedProx slightly outperform in final ACC (above 0.98), FL-IRT shows strong F1 and MCC scores with 24 clients, reaching about 0.97 for the ACC indicator and 0.96 for the MCC one. With fewer clients, FL-IRT remains competitive but exhibits higher standard deviation values, particularly in the 6-client case due to the limited representation of all classes.

Table 7.6: MNIST dataset: evaluation metrics averaged over the last 10 communication rounds. Values are reported as mean ± standard deviation.

Method	**6 Clients**			**12 Clients**			**24 Clients**		
	Acc	**F1-Score**	**MCC**	**Acc**	**F1-Score**	**MCC**	**Acc**	**F1-Score**	**MCC**
FL-IRT	0.880 ± 0.085	0.856 ± 0.115	0.857 ± 0.095	0.948 ± 0.047	0.945 ± 0.056	0.934 ± 0.056	0.970 ± 0.011	0.970 ± 0.010	0.960 ± 0.014
FedAvg	0.976 ± 0.002	0.976 ± 0.002	0.968 ± 0.002	0.980 ± 0.001	0.980 ± 0.001	0.974 ± 0.001	0.980 ± 0.000	0.980 ± 0.000	0.974 ± 0.001
FedProx	0.980 ± 0.001	0.979 ± 0.001	0.973 ± 0.002	0.980 ± 0.001	0.979 ± 0.001	0.973 ± 0.001	0.982 ± 0.001	0.981 ± 0.001	0.976 ± 0.002

Fig. 7.6 shows performance over communication rounds for the 24 clients scenario. FL-IRT maintains low variations and tracks closely behind FedAvg and FedProx, with marginal differences in final scores. Despite not reaching the absolute peak ACC, it offers interpretable models with reasonable stability.

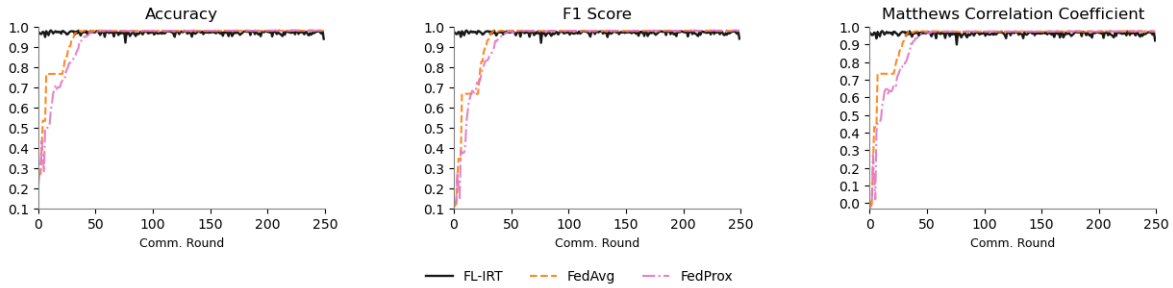


Figure 7.6: MNIST dataset: test metrics across all communication rounds for the 24-client configuration.

Finally, Tab. 7.7 demonstrates the runtime efficiency of the proposed FL-IRT method. Across all client configurations, FL-IRT trains between 21× and 26× times faster than FedAvg, and between 83× and 239× times faster than FedProx. These substantial reductions in processing time highlight the method's suitability for efficient and scalable deployment in real-world FL scenarios.

Table 7.7: MNIST dataset: processing time (in seconds) for different numbers of clients.

Method	**6 Clients (s)**	**12 Clients (s)**	**24 Clients (s)**
FL-IRT	103.0	174.1	301.2
FedAvg	2086.4	4161.0	7805.4
FedProx	8494.9	41577.3	66795.6

7.3.6.4 Adult Dataset

The ADL dataset, though IID across clients, presents classification challenges due to severe class imbalance. As shown in Tab. 7.8, DNN-based methods, i.e. FedAvg, FedProx, and HOLDA, consistently achieve strong results across ACC, F1-score, and MCC. HOLDA metrics. In particular, ACC achieves an ACC value of 0.829 while the MCC indicator reaches about 0.50, indicating better handling of the minority class compared to other models.

FL-IRT offers a competitive alternative w.r.t. other rule-based methods while ensuring full interpretability. In fact, it achieves an ACC value about 0.78, a F1-score about 0.74, and a MCC value about 0.30. Although it lags slightly behind DNN-based methods, FL-IRT maintains full interpretability without relying on surrogate models or post-hoc explanations. Its performance remains stable and sufficiently accurate under class-imbalanced conditions.

The rule-based methods G-Explainer and S-TREPAN achieve acceptable F1-scores but yield significantly lower MCC values, reflecting poor performance on imbalanced class distributions. Their global DTs, extracted from opaque neural models, lack the reliability and transparency offered by FL-IRT's native rule-based description.

Table 7.8: *Adult* dataset: evaluation metrics averaged over the last 10 communication rounds (4 clients). Values are reported as mean ± standard deviation.

Method	**Acc**	**F1-Score**	**MCC**
FL-IRT	0.777 ± 0.009	0.744 ± 0.020	0.297 ± 0.056
FedAvg	0.821 ± 0.001	0.751 ± 0.005	0.516 ± 0.005
FedProx	0.820 ± 0.003	0.752 ± 0.008	0.515 ± 0.010
HOLDA	0.829 ± 0.024	0.811 ± 0.051	0.503 ± 0.114
G-Explainer	0.797 ± 0.021	0.781 ± 0.023	0.426 ± 0.040
S-TREPAN	0.785 ± 0.008	0.746 ± 0.019	0.339 ± 0.030

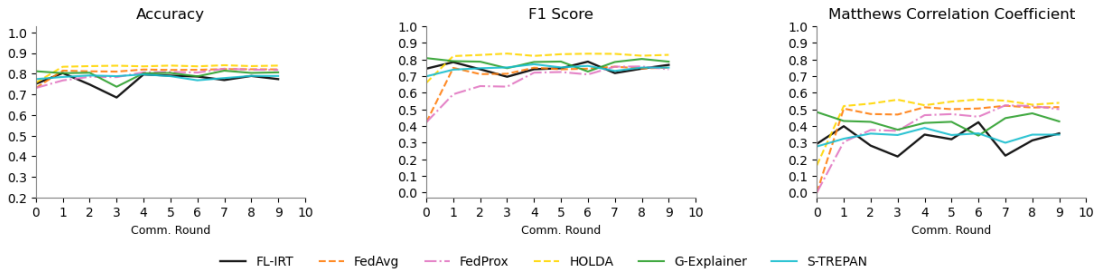


Figure 7.7: ADL dataset: test metrics across all communication rounds for the 4-client configuration.

Fig. 7.7 illustrates the trends over communication rounds. FL-IRT converges early and shows moderate variations across metrics. HOLDA performs consistently well but with wider confidence intervals. G-Explainer and S-TREPAN display fluctuating behavior, particularly in MCC.

Finally, Tab. 7.9 highlights runtime performance. FL-IRT completes training and explanation in 111 s, about 10% faster than FedAvg (124 s), and about 94% faster than FedProx (1813 s). Notably, rule-based methods G-Explainer and S-TREPAN require over 237% more time than FL-IRT, despite leveraging GPU acceleration, whereas FL-IRT runs entirely on CPU.

Table 7.9: *Adult* dataset: processing time (in seconds) over 10 communication rounds (4 clients).

Method	**Processing Time (s)**
FL-IRT	111.0
FedAvg	124.0
FedProx	1813.0
HOLDA	158.7
G-Explainer	369.2
S-TREPAN	374.7

7.3.6.5 Explainability through Native Rule Reasoning

A central objective of FL-IRT is to ensure full transparency throughout the FL pipeline. Unlike neural-based methods such as FedAvg and FedProx, that produce opaque weight matrices and depend on black-box aggregation, FL-IRT builds interpretable logic directly, at both client and server levels, through native DT-based rule extraction.

Client-Level Interpretability Each client trains a DT locally, and the branching logic is directly converted into a symbolic rule set. These rules consist of interpretable threshold conditions (e.g., on X_1, X_2), providing domain experts with human-readable insights into the model's decisions.

Server-Side Reasoning and Global Fusion The server collects the rule outputs and constructs a synthetic dataset for global training. A new interpretable DT is then trained on this data, capturing global logic that reconciles conflicting or incomplete local patterns.

Global Feature Recovery Under Non-IID Settings In the HSP dataset, clients from different groups learned rules solely based on X_1 and X_2, omitting the critical feature *Age*. However, the server's global model inferred that *age* is a discriminative variable for reconciling these group behaviors, thereby integrating it into the final rule set.

Limitations of Post-Hoc Explainability This feature-level insight is not recovered by post-hoc rule extraction frameworks such as GLOR-FLEX. For instance, the G-Explainer component fails to include *age* in the final rule set because it is absent from all local rules. In contrast, FL-IRT preserves full feature awareness during synthetic labeling and training, enabling richer and more accurate global explanations.

As illustrated in Tab. 7.10, these representative global rules demonstrate how FL-IRT reconciles divergent local logic and incorporates *Age*, even when it is absent from all client rules. This highlights FL-IRT's ability to go beyond localized logic and synthesize globally coherent and transparent decision rules—even when key features are underutilized locally.

Table 7.10: Illustrative local and global rules showing recovery of *Age* by FL-IRT (HSP dataset).

Source	Extracted Rules	Interpretation
Client Group 1	**if** ($X_1 > 4.992$) **then** **Unhealthy** **if** ($X_2 > 5.067$) **then** **Unhealthy** **if** ($X_2 \leq 5.057$) **then** **Healthy**	Local logic only uses X_1 and X_2; *Age* is absent.
Client Group 2	**if** ($X_1 \leq 5.681$) **then** **Unhealthy** **if** ($X_1 > 4.403$) **then** **Healthy** **if** ($X_1 \leq 5.244$) **then** **Unhealthy**	Alternate thresholds on X_1; *Age* still not present.
Server (FL-IRT Global)	**if** ($X_1 > 5.242$ **and** $X_2 > 5.064$ **and** *Age* > 45.0) **then** **Healthy** **if** ($X_1 \leq 5.242$ **and** *Age* ≤ 45.0 **and** $X_2 > 5.068$) **then** **Unhealthy** **if** ($X_1 \leq 5.242$ **and** *Age* > 45.0) **then** **Unhealthy**	Server incorporates *Age* to reconcile inconsistencies between client groups.

7.3.6.6 Computational Overhead and ACC degradation due to Privacy procedures

Tab. 7.11 reports the computational overhead observed when enabling TLS-based secure communication and DP mechanisms in FL-IRT, specifically using the VP dataset in the scenario with 24 clients.

In the baseline FL-IRT configuration without any privacy mechanisms, the total runtime was 545.3 s. When both TLS and DP are enabled in the complete FL-IRT framework, the total runtime increases to 5090.4 s, resulting in an additional 4545.07 s of TLS/DP overhead, equivalent to an $\sim$ 833% increase over the baseline.

Among the privacy-related components, TLS communication alone contributes for 1904.9 s, accounting for $\sim$ 37% of the total runtime of the complete FL-IRT model. The remaining overhead ($\sim$ 2640 s or $\sim$ 52%) is primarily attributed to per-rule DP processing, including score obfuscation and potential rule filtering or clipping logic.

However, the application of the DP procedures results in a reduction of the performance metrics. Using the VP dataset and the 24-client configuration as a representative example, Tab. 7.12 reports degradations of approximately 9% for ACC, 10% for F1-score, and 19% for MCC. Comparable results are observed for the other datasets: with the selected $\epsilon = 0.5$

parameter, an ACC reduction of about 5% to 10% can be expected, while the F1-score and MCC metrics are typically more affected.

Despite the significant increase in runtime and some performance degradation, both TLS and DP mechanisms provide strong guarantees for secure transmission and local rule privacy. However, it is worth noticing that, if needed, only TLS procedures can be applied to the baseline FL-IRT model to protect the communication channel at the cost of a reduction in the privacy protection. Therefore, the application of the DP procedures can be included in the FL-IRT processing chain, or not, depending on the specific scenario and on the probability of a malicious attack [196, 197].

These examples highlight the computational trade-off introduced by privacy-preserving procedures in FL systems and quantify their impact in practical deployments.

Table 7.11: VP dataset (24 clients): computational overhead introduced by TLS and DP mechanisms in FL-IRT.

Metric	**Value**
Avg. TLS time/round (ms)	7619.5
Max. TLS time/round (ms)	8774.3
Total TLS time (s)	1904.9
Total DP time (s)	2640.2
Complete FL-IRT time (s)	5090.4
TLS/DP vs complete time ratio (%)	~ 89
Baseline FL-IRT time (s)	545.3
Complete vs baseline time ratio (%)	~ 833

Table 7.12: VP dataset (24 clients): effects of the DP procedure on the ACC of the baseline vs the complete FL-IRT framework. ACC values are averaged over the last 10 communication rounds and reported as mean only.

Method	**ACC**	**F1-Score**	**MCC**
Baseline FL-IRT	0.856	0.856	0.713
Complete FL-IRT	0.782	0.769	0.576
ACC reduction (%)	~ 9	~ 10	~ 19

7.3.6.7 Decision Tree convergence behavior

Fig. 7.8 deals with the convergence properties of the DT method with the VP dataset, by measuring the asymptotic trend of False Positive Rate (FPR) (top left) and False Negative Rate (FNR) (top right), under an increasing number of samples of the training

dataset. Likewise, Fig. 7.8 shows the corresponding average ACC (bottom left) and the misclassification rate (bottom right) for the same dataset. It is apparent that the flat trends start at about 3000 samples. It is worth noticing that the results shown in Fig. 7.5 correspond to 4800 samples for the configuration with 24 clients.

In addition, as already discussed in Sect. 7.3.6.2 and sketched in Fig. 7.5, the DNN approaches require more communication rounds to achieve good steady performance. The rationale behind such a slow convergence relies on their more complex structure. The fast DT convergence is apparent here for the VP dataset. However, this is not always true in general. In fact, more noisy datasets may require more complex decision boundaries between the classes, thus increasing the DT architectural complexity and slowing down the DT adaptation rate. Further considerations about the architectural complexity of the methods and the impact of different datasets are given in the next Sect. 7.3.6.8 while, especially for the DT method, some results are summarized in Tab. 7.16. Further investigation on this topic is left open for future research activities.

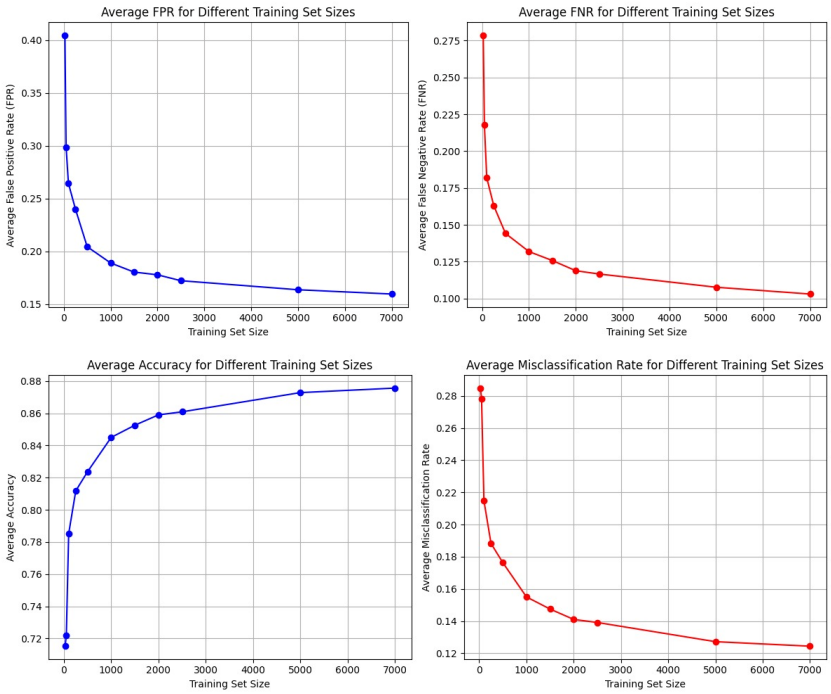


Figure 7.8: Average FPR (top left), FNR (top right), ACC (bottom left), and misclassification rate (bottom right) of the DT method for an increasing number of samples of the training VP dataset.

7.3.6.8 Architectural complexity and memory footprint

According to the DNN architectures employed in the simulations for the FedAvg, FedProx, and HOLDA methods described in Sect. 7.3.5, the number of parameters and memory requirements (expressed in bytes) are summarized in Tab. 7.13. For the memory requirements, the single-precision floating-point format (i.e., the `float32` type, occupying 4 bytes in memory) is assumed for all parameters. Since GLOR-FLEX-derived methods cannot handle multi-class datasets, the term *Not Available* (N/A) is used to indicate that HOLDA cannot manage the MNIST dataset.

Table 7.13: DNN size metrics (total parameters and memory in bytes) for FedAvg/FedProx and HOLDA across datasets.

Dataset	**FedAvg / FedProx**		**HOLDA**	
	Parameters	**Memory (Bytes)**	**Parameters**	**Memory (Bytes)**
HSP	3735	14940	67330	269320
VP	8395	33580	68354	273416
MNIST	3814	15256	N/A	N/A
ADL	7696	30784	70146	280584

Tabs. 7.14 and 7.15 present the memory footprint and structural metrics for the G-Explainer and S-TREPAN models, respectively. G-Explainer consistently yields the most compact models, where the G-Explainer average sizes remaining well below 40kB even in complex settings such as the VP dataset under the 24-client configuration. However, its reliance on post-hoc aggregation of distributed rules limits its flexibility and occasionally omits critical predictive features. S-TREPAN, on the other hand, maintains extremely stable tree structures across client counts, especially in the VP dataset, where the average Tree Size (TS) and depth (TD) remain fixed, thus suggesting limited responsiveness to new data. While this stability reduces variations and memory load, it can hinder expressiveness and adaptation in heterogeneous environments.

On the contrary, Tab. 7.16 presents the metrics concerning tree-related sizes across different datasets and client scenarios for the FL-IRT framework. These metrics include Tree Size (TS), Tree Depth (TD), Number of Internal Nodes (N_{IN}), Number of Leaf Nodes (N_{LN}), and Model Size (MS).

The results show how the architectural complexity of FL-IRT evolves across datasets and client configurations. In the HSP dataset, increasing client participation leads to smaller and more efficient trees since the average TS decreases from $\sim$ 55 (6 clients) to just over 31 (for 12 and 24 clients), with a similar reduction in depth and model size. This

Table 7.14: G-Explainer model size (average and standard deviation) in bytes across datasets and client configurations.

Dataset	Clients	Size	Min	Max
HSP	6	1351 ± 65	976	1792
	12	2366 ± 129	1512	2944
	24	4456 ± 391	2488	5504
VP	6	7887 ± 2315	248	14600
	12	16914 ± 3120	7720	25072
	24	36328 ± 4528	24032	55704
ADL	4	1234 ± 947	632	3888

Table 7.15: S-TREPAN tree and model size metrics for different datasets and client counts. TS = Tree Size, TD = Tree Depth, N_{IN} = Number of Internal Nodes, N_{LN} = Number of Leaf Nodes, and MS = Model Size (Bytes).

Metric	HSP (Clients)			VP (Clients)			MNIST (Clients)	ADL (Clients)
	6	12	24	6	12	24	N/A	4
Avg. TS	6.18	10.95	10.04	29.0	29.0	29.0	N/A	10.2
Min. TS	5	9	7	29	29	29	N/A	7
Max. TS	21	29	13	29	29	29	N/A	11
Avg. TD	3.52	4.80	4.50	13.19	13.91	14.62	N/A	4.7
Min. TD	3	4	3	8	9	11	N/A	4
Max. TD	7	8	6	15	15	15	N/A	5
Avg. N_{IN}	2.59	4.98	4.52	14.0	14.0	14.0	N/A	4.6
Min. N_{IN}	2	4	3	14	14	14	N/A	3
Max. N_{IN}	10	14	6	14	14	14	N/A	5
Avg. N_{LN}	3.59	5.98	5.52	15.0	15.0	15.0	N/A	5.6
Min. N_{LN}	3	5	4	15	15	15	N/A	4
Max. N_{LN}	11	15	7	15	15	15	N/A	6
Avg. MS	5623.3	9247.3	8600.3	23752.6	23813.8	23843.4	N/A	8656.0
Min. MS	4728	7784	6256	23608	23704	23736	N/A	6160
Max. MS	16696	22680	10840	23864	23864	23864	N/A	9280

pattern suggests that as more data diversity is aggregated, the server can generalize more effectively, reducing redundancy in the global rule set. In contrast, the VP dataset exhibits consistently large trees (TS $\sim$ 302–305) and deep structures (TD $\sim$ 14–15), regardless of client count. This reflects the dataset's higher feature complexity and dense rule generation, which demand greater model expressiveness to maintain performance.

The MNIST dataset shows a growth in average TS from $\sim$ 83 (6 clients) to $\sim$ 205 (24 clients), aligned with its multi-class and non-IID structure, where increased participation introduces more variation to be encoded. The Adult dataset, by comparison, exhibits moderate complexity: with just 4 clients (see Sect. 7.3.5), FL-IRT produces a relatively compact tree (TS = 41, MS $\sim$ 3kB), confirming its suitability for imbalanced binary classification under constrained federation. Overall, the bolded TS and MS values underscore FL-IRT's scalability from lightweight models in simple settings to robust, high-capacity trees for complex or multi-client scenarios, while remaining interpretable and resource-efficient throughout.

Table 7.16: Tree and model size metrics for different datasets and client counts. TS = Tree Size, TD = Tree Depth, N_{IN} = Number of Internal Nodes, N_{LN} = Number of Leaf Nodes, and MS = Model Size (Bytes).

Metric	**HSP (Clients)**			**VP (Clients)**			**MNIST (Clients)**			**ADL (Clients)**
	6	12	24	6	12	24	6	12	24	4
Avg. TS	**55.2**	**31.2**	**31.6**	**304.7**	**302.1**	**304.1**	**83.4**	**145.7**	**204.9**	**41.0**
Min. TS	11	15	15	245	239	217	33	49	63	13
Max. TS	459	157	209	367	377	427	185	311	379	85
Avg. TD	8.0	6.2	6.3	14.3	14.3	15.06	10.0	12.0	13.5	8.8
Min. TD	3	3	4	11	11	11	6	8	9	4
Max. TD	459	157	209	21	20	23	15	17	20	17
Avg. N_{IN}	27.1	15.1	15.3	151.9	150.5	151.6	41.2	72.3	102.0	20.0
Min. N_{IN}	5	7	7	122	119	108	16	24	31	6
Max. N_{IN}	229	78	104	183	188	213	92	155	189	42
Avg. N_{LN}	28.1	16.1	16.3	152.9	151.5	152.6	42.2	73.3	103.0	21.0
Min. N_{LN}	6	8	8	123	120	109	17	25	32	7
Max. N_{LN}	230	79	105	184	189	214	93	156	190	43
Avg. MS	3799.4	2839.8	2853.9	**13781.4**	**13674.5**	**13757.3**	5611.2	8599.5	11443.3	3029.4
Min. MS	2018	2040	2040	2040	2040	2040	2056	2056	2056	2004
Max. MS	37864	13701	17861	30504	31304	35304	18917	31016	37544	7191

Compared to both G-Explainer and S-TREPAN methods, FL-IRT offers a more adaptive and integrated framework to global model construction. While G-Explainer is lightweight, it can decouple rule extraction from training even if it may suffer from rule fragmentation and incompleteness. S-TREPAN provides stability but lacks adaptive scaling in complex domains. On the contrary, FL-IRT directly incorporates client diversity and local updates into the construction of a unified, explainable decision structure,

achieving a favorable trade-off between ACC, interpretability, and memory cost. These results support FL-IRT as a robust middle ground between compactness and adaptability in federated explainable modeling.

7.4 Conclusion

This chapter investigated the integration of explainability into Federated Learning (FL) systems, introducing Federated Learning for Interpretable Rule Trees (FL-IRT) as a rule-based alternative to opaque deep learning approaches. Unlike conventional frameworks such as Federated Averaging (FedAvg) and Federated Proximal optimization (FedProx), which rely on neural weight aggregation and post-hoc interpretation, FL-IRT embeds interpretability natively by constructing symbolic rules at both the client and server levels. The framework combines client-side Decision Tree (DT) learning with server-side synthetic data labeling, enabling a global, interpretable model that maintains robustness under heterogeneous conditions. Privacy-preserving mechanisms, including Transport Layer Security (TLS) encryption and Differential Privacy (DP), were also incorporated to ensure secure communication and data protection in distributed settings.

The evaluation compared FL-IRT against FedAvg, FedProx, and three models derived from the Global-Local Rule-Based eXplanations for Federated Learning (GLOR-FLEX) framework, namely Hierarchical crOss-siLo feDerated Averaging (HOLDA), Global-Explainer (G-Explainer), and Server-TREPAN (S-TREPAN). Four benchmark datasets were employed, covering diverse statistical properties and federated scenarios: the Hospital dataset (HSP), the Vehicle Platooning dataset (VP), the Modified National Institute of Standards and Technology dataset (MNIST), and the Adult dataset (ADL).

On the HSP dataset, characterized by strong non-Independent and Identically Distributed (non-IID) conditions and label inversion across groups, FL-IRT achieved near-perfect performance with Accuracy (ACC) values approaching 0.999 and high Matthews Correlation Coefficient (MCC) scores. It outperformed both neural baselines and GLOR-FLEX-derived methods, successfully recovering globally discriminative features such as *Age* even when absent in local models. This demonstrated the ability of FL-IRT to reconcile divergent local behaviors into coherent and interpretable global rules.

On the VP dataset, FL-IRT maintained ACC in the 0.85 range, closely matching FedAvg and FedProx while offering substantially lower runtime. Although FedAvg attained slightly higher peak ACC, it converged more slowly and exhibited greater instability

under heterogeneous conditions. By contrast, FL-IRT converged earlier with lower variance, providing a stable and interpretable alternative. GLOR-FLEX models, particularly G-Explainer, suffered from volatility and degraded generalization, while S-TREPAN exhibited stability at the cost of reduced flexibility.

The MNIST dataset further emphasized the adaptability of FL-IRT. Despite the complexity introduced by disjoint digit distributions across client groups, FL-IRT reached accuracies of approximately 0.97 and MCC values near 0.96 with 24 clients. While FedAvg and FedProx attained slightly higher peak ACCs above 0.98, they required considerably longer runtimes and yielded opaque models. Importantly, GLOR-FLEX-derived methods could not handle the multi-class setting, underscoring the broader applicability of FL-IRT. Runtime analysis showed speedups of up to 200× relative to FedProx, confirming FL-IRT's scalability.

On the ADL dataset, where class imbalance complicates decision boundaries, FL-IRT sustained ACC near 0.78 and F1-scores around 0.74. While slightly behind FedAvg, FedProx, and HOLDA in predictive performance, it consistently outperformed G-Explainer and S-TREPAN, avoiding the volatility of post-hoc explainers while maintaining native interpretability.

Beyond predictive metrics, this chapter assessed computational efficiency, convergence behavior, and memory footprint. Across all datasets, FL-IRT achieved runtime reductions between 70% and 95% compared to deep learning baselines, even when executed entirely on CPU. Its symbolic models converged faster and required fewer resources than Deep Neural Networks (DNNs). Privacy-preserving mechanisms introduced noticeable overhead, with TLS and DP increasing runtime by up to eightfold and reducing accuracy by 5–10%, but they ensured secure communication and rule confidentiality.

In summary, FL-IRT emerges as a practical and principled framework for explainable FL. It consistently balances interpretability, computational efficiency, and robustness under heterogeneous conditions. Compared to FedAvg and FedProx, it delivers competitive performance with far greater transparency, while surpassing GLOR-FLEX-derived approaches in both stability and generalization. By embedding explainability directly within the federated pipeline rather than appending it post-hoc, FL-IRT provides a trustworthy and efficient solution suitable for safety-critical and regulated domains. These findings establish FL-IRT as a robust step toward reconciling the demands of privacy, interpretability, and predictive accuracy in distributed learning environments.

Acknowledgements This work was supported by the following projects:

REXASI-PRO Horizon Europe project, call HORIZON-CL4-HUMAN-01-01, Grant Agreement ID: 101070028.

Future Artificial Intelligence Research (FAIR) project, Italian National Recovery and Resilience Plan (PNRR), Spoke 3 – ResilientAI.

STRIVE (la Scienza per le TRansizioni Industriale, Verde, Energetica) FOE MUR 2022–2025, subproject MeSaS.

CHAPTER 8

General conclusion

This thesis has explored the integration of eXplainable Artificial Intelligence (XAI) into Artificial Intelligence (AI) pipelines for safety–critical applications. It had a two-fold focus: (i) perception for autonomous and assistive mobility, and (ii) distributed learning under privacy and ownership constraints. Therefore, the central contention here has been that explainability should not be regarded as a peripheral feature appended after training but, rather, as a core design principle governing how systems should be constructed, validated, and deployed. The studies conducted across perception monitoring—including rule-based Operational design domain (ODD) evaluation, scene-level reasoning with Vision-Language Models (VLMs), and prediction certainty through Uncertainty Quantification (UQ)—as well as Federated Learning (FL) demonstrate that transparency can be embedded alongside high–performance models, thereby improving robustness, accountability, and regulatory alignment.

This thesis set strong baselines from a perception viewpoint by training high–performance object detectors targeted at assistive indoor mobility. The fine–tuned YOLOv8s model reached an mAP@0.5 of 0.97 for person detection and 0.99 for wheelchair detection on the independent CNR Villa dataset, which corroborates the robustness of the predictive backbone. Building on that, explainability was integrated at various levels. Feature–based analysis using Cumulative Feature Ranking (CFR) identified key visual attributes responsible for the detection robustness. Interpretable rule–based classifiers and Decision Trees (DTs) formalized Operational design domain (ODD) boundaries providing lightweight but auditable safety checks. Vision-Language Model (VLM) extended this to semantic scene–level interpretation, transforming raw sensor input into natural–language risk assessment. Finally, Conformal Prediction (CP) introduced statistical guarantees, which converted single predictions into calibrated sets with explicit error control. Together, these approaches have shown how perception can be simultaneously made accurate, interpretable, and accountable in safety–critical contexts

such as Autonomous Driving Systems (ADS), Advanced Driver-Assistance Systems (ADAS), and assistive mobility platforms.

A central insight is that, although each perception-related contribution was evaluated independently, together they illustrate a layered methodology for explainable perception. At the foundational level, detectors must achieve both high predictive performance and structural interpretability. At the intermediate level, prediction reliability should be rigorously quantified through distribution-free uncertainty estimation methods such as CP, and expressed in a form usable by downstream modules. At the highest level, semantic reasoning can help resolve the ambiguities of complex environments, including gray-swan and black-swan risks. This multi-layer perspective conceptually shows how explainability may extend across abstraction levels—from pixel-level features through rule-based ODD compliance up to semantic risk reasoning—offering a coherent framework for accountability in safety-critical perception systems.

In the field of distributed training, this thesis reframed FL as a process not only of optimization but of knowledge governance. Conventional FL methods aggregate opaque Deep Neural Networks (DNNs), which limits transparency, auditability, and regulatory compliance. In this respect, this work introduced the Federated Learning for Interpretable Rule Trees (FL-IRT) framework, whereby interpretable artifacts—essentially rule–based surrogates, such as decision trees extracted locally—are combined into a global representation. This immediately shifts the focus from black–box parameter averaging to constructing interpretable structures that retain accuracy comparable to traditional FL while revealing their underlying decision logic. What is more, the incorporation of secure communication mechanisms, such as Transport Layer Security (TLS), along with adherence to General Data Protection Regulation (GDPR) principles, has shown that both privacy preservation and interpretability can be achieved jointly. These contributions together position FL-IRT as an interpretable alternative to standard FL, offering a distributed learning paradigm that is simultaneously performant, transparent, and compatible with safety and regulatory requirements.

The broader significance of these findings stands in the unifying principle they reveal: explainability forms a regulative layer in AI, operationalizing principles of safety, accountability, and human oversight. It does so in perception by regularizing the mapping from sensor inputs to safety–critical decisions via interpretable justifications and calibrated uncertainty bounds. It does so in federated training by controlling how information flows across distributed agents, restricting global aggregation towards human–understandable and auditable structures. In all cases, explainability acts to govern what systems may

assert, how such assertions might be justified, and how their failures can be verified. This reframes XAI from the realm of visualization or post–hoc interpretation into the domain of system design and accountability.

From a scientific perspective, three general conclusions emerge. First, reliability in safety–critical Cyber-Physical Systems (CPSs) must be multidimensional: combining predictive Accuracy (ACC) with principled UQ and operational interpretability. Second, explainability needs to be integrated at multiple layers of abstraction—from feature–level attribution to rule–based reasoning and statistical certification—each supporting a distinct dimension of safety assurance. Third, governance in AI can be made measurable through quantitative assessment of uncertainty margins, interpretability costs, and communication overhead, hence transforming abstract ethical and legal principles into concrete engineering standards. In this regard, it is shown that explainability can be infused into perception and distributed learning pipelines without excessively sacrificing accuracy or efficiency. The developed methodologies lay the foundation for constructing AI systems that achieve performance, trust, and certifiability. The broader implication is that XAI should no longer be treated as an afterthought but as a central design principle—guiding the construction of systems that are accountable, auditable, and aligned with human and regulatory expectations.

Appendix

Acknowledgements

I would like to express my deepest gratitude to my family, especially my parents, for their continuous love and support, even from afar during these years. To my partner, Behshad, I am profoundly thankful for being by my side through every step of this journey.

My most sincere and heartfelt thanks go to my co-supervisor, Maurizio Mongelli. His guidance and expertise shaped my research throughout these years, while his constant support went far beyond academic matters. He stood by my side in difficult times, and his help was crucial in solving many personal and administrative issues, including my residence permit. Without his mentorship and encouragement, this thesis would not have been possible.

I am also grateful to the researchers at the National Research Council (CNR), in particular Vittorio Rampa and Alessia Paglialonga, for their valuable support and collaboration.

My sincere appreciation goes to my supervisor, Fabio Patrone, for his guidance throughout my doctoral studies. From the University of Genoa, I would also like to thank Professor Mario Marchese for his assistance in resolving critical legal and residence-related matters.

From Aitek, I am truly grateful to my industrial co-advisor, Stefano Delucchi, as well as my colleagues Vanessa Orani, Jacopo Motta, and Alberto Carlevaro, whose support and collaboration greatly contributed to shaping the direction of my research.

Finally, I would like to express my gratitude to Professor Hendro Wicaksono from Constructor University, Bremen, for giving me the opportunity to have a highly productive and enriching international research experience during my stay abroad.

Publication record

1. International Journal Papers

[1] A. Samandari, V. Rampa, M. Marchese, A. Paglialonga, F. Patrone, and M. Mongelli, "Federated Learning for Rule-Based Systems," submitted to *ACM Transactions on Intelligent Systems and Technology (TIST)*, September 2025.

2. International Conference Papers

[2] A. Samandari, M. Marchese, A. Paglialonga, F. Patrone, V. Rampa, and M. Mongelli, "Federated Learning for Rule-Based Systems: Preliminary Studies," *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*, pp. 143–148, Sep. 2024, IEEE. doi: 10.1109/RTSI60181.2024.10704658.

[3] A. Samandari, J. Motta, V. Orani, A. Carlevaro, and M. Mongelli, "eXplainable Checker of Video Analytics Performance in Indoor Smart Mobility," in *Intelligent Systems Conference (IntelliSys 2025)*, K. Arai (Ed.), Lecture Notes in Networks and Systems, vol. 1554, pp. 1–20, Springer, 2025. doi: 10.1007/978-3-031-99965-9_24.

Participation in Research Projects

REXASI-PRO – REliable & eXplainable Swarm Intelligence for People with Reduced mObility
Funding body: European Union – Horizon Europe

Short project description: European project focused on developing autonomous mobility solutions for people with reduced mobility through reliable and explainable AI.

Activity: Contributed to the development of the autonomous wheelchair perception and monitoring module. Activities included object detection model tuning, integration of explainable AI for video analytics, semantic safety assessment using Vision-Language Models, and uncertainty quantification with conformal prediction to enhance system transparency and reliability.

Experience outside the University of Genoa during the PhD

Consiglio Nazionale delle Ricerche (CNR), Genoa, Italy
Period: 2022–2025 (affiliated during PhD)
Short activity description: Active involvement in the EU project REXASI-PRO, focusing on explainability in video analytics for autonomous systems. In collaboration with CNR researchers, contributed to the design of explainable perception modules and co-developed the FL-IRT framework for interpretable federated learning.

Aitek S.p.A., Genoa, Italy
Period: 2022–2025 (approx., twice-weekly presence)
Short activity description: Contributed to REXASI-PRO activities on perception modules, with a focus on developing and fine-tuning object detection models. Actively collaborated with industrial researchers, bridging academic research with applied deployment scenarios.

Constructor University, Bremen, Germany
Period: 6 months (March–August 2025)
Short activity description: Collaborative research on causal discovery and federated learning. Jointly developed a prototype causal federated learning system based on synthetic data sharing, leading to successful results and preparation of a journal manuscript.

Attendance to workshops

2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI), Milan, Italy, 18–20 Sep. 2024. Presenting: A. Samandari, M. Marchese, A. Paglialonga, F. Patrone, V. Rampa, and M. Mongelli, "Federated Learning for Rule-Based Systems: Preliminary Studies," included in proceedings.

Intelligent Systems Conference (IntelliSys 2025), Amsterdam, The Netherlands, 28–29 Aug. 2025. Presenting: A. Samandari, J. Motta, V. Orani, A. Carlevaro, and M. Mongelli, "eXplainable Checker of Video Analytics Performance in Indoor Smart Mobility," included in proceedings (Springer LNNS).

Teaching and other activities

Co-supervision of Bachelor Thesis: "Causal Discovery in Federated Learning Using Synthetic Data: A Comparative Analysis of Generation Methods and Graph Fidelity" by Matilda Milisavljevic, Constructor University, Bremen, Germany, 2024.

Seminar speaker on explainability in autonomous systems and federated learning.

Reviewing activity for the following international journals and conferences:

> *Engineering Applications of Artificial Intelligence (EAAI)* – Reviewed paper Ref: EAAI-25-1681, titled "Exploring the Determinants of Accident Fatalities: A Causal Analysis of Urban-Rural Disparities Using Double Machine Learning" (2025).
>
> *xAI-2025 Conference (main track)* – Reviewed paper ID 7904, titled "From Input to Insight: Probing the Reasoning of Attention-based MIL Models" (2025).
>
> *xAI-2024 Conference (main track)* – Reviewed paper ID 5653, titled "Enhancing Trustworthiness through the Optimization of Explainable AI for the Safety-Critical Use Case of Runway Detection" (2024).

Visiting student at Constructor University, Bremen, Germany (2025).

Bibliography

[1] NVIDIA. (2025) What are vision-language models (vlms)? https://www.nvidia.com/en-us/glossary/vision-language-models/. Accessed: 2025-09-09.

[2] X. Zhou, B. Chen, Y. Gui, and L. Cheng, "Conformal prediction: A data perspective," *ACM Comput. Surv.*, vol. 58, no. 2, Sep. 2025. [Online]. Available: https://doi.org/10.1145/3736575

[3] L. Andéol, T. Fel, F. De Grancey, and L. Mossina, "Confident object detection via conformal prediction and conformal risk control: an application to railway signaling," in *Conformal and Probabilistic Prediction with Applications*. PMLR, 2023, pp. 36–55.

[4] R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan *et al.*, "Explainable ai (xai): Core ideas, techniques, and solutions," *ACM computing surveys*, vol. 55, no. 9, pp. 1–33, 2023.

[5] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[7] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[8] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[9] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley, "Explainable machine learning in deployment," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 648–657.

[10] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *Journal of Machine Learning Research*, vol. 20, no. 177, pp. 1–81, 2019.

[11] J. Friedman and B. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, 12 2008.

[12] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[13] B. Greenwell, "pdp: An r package for constructing partial dependence plots," *R Journal*, vol. 9, pp. 421–436, 06 2017.

[14] D. Lewis, *Counterfactuals*. John Wiley & Sons, 2013.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[16] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[17] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," *ArXiv*, vol. abs/1904.12584, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:108296442

[18] D. Doran, S. Schulz, and T. R. Besold, "What does explainable AI really mean? A new conceptualization of perspectives," in *Proceedings of the*

*First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017*, ser. CEUR Workshop Proceedings, T. R. Besold and O. Kutz, Eds., vol. 2071. CEUR-WS.org, 2017. [Online]. Available: https://ceur-ws.org/Vol-2071/CExAIIA_2017_paper_2.pdf

[19] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," *SAE Standard J*, vol. 3016, p. 1, 2014.

[20] J. Kocić, N. Jovičić, and V. Drndarević, "Sensors and sensor fusion in autonomous vehicles," in *2018 26th Telecommunications Forum (TELFOR)*. IEEE, 2018, pp. 420–425.

[21] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[22] X. Li, K.-Y. Lin, M. Meng, X. Li, L. Li, Y. Hong, and J. Chen, "A survey of adas perceptions with development in china," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 188–14 203, 2022.

[23] J. Janai, F. Güney, A. Behl, A. Geiger *et al.*, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," *Foundations and trends® in computer graphics and vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[24] K. Bayoudh, F. Hamdaoui, and A. Mtibaa, "Transfer learning based hybrid 2d-3d cnn for traffic sign recognition and semantic road detection applied in advanced driver assistance systems," *Applied Intelligence*, vol. 51, no. 1, pp. 124–142, 2021.

[25] J. H. Kim, G. Batchuluun, and K. R. Park, "Pedestrian detection based on faster r-cnn in nighttime by fusing deep convolutional features of successive images," *Expert Systems with Applications*, vol. 114, pp. 15–33, 2018.

[26] G. Li, Y. Yang, and X. Qu, "Deep learning approaches on pedestrian detection in hazy weather," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 10, pp. 8889–8899, 2019.

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[28] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[29] S. Guo, S. Wang, Z. Yang, L. Wang, H. Zhang, P. Guo, Y. Gao, and J. Guo, "A review of deep learning-based visual multi-object tracking algorithms for autonomous driving," *Applied Sciences*, vol. 12, no. 21, p. 10741, 2022.

[30] J. Van Brummelen, M. O'brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transportation research part C: emerging technologies*, vol. 89, pp. 384–406, 2018.

[31] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Fort Lauderdale, Florida, USA: PMLR, 2017, pp. 1273–1282.

[33] P. M. Mammen, "Federated learning: Opportunities and challenges," *arXiv preprint arXiv:2101.05428*, 2021.

[34] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[35] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[36] 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.

[37] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[38] 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.

[39] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[40] 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.

[41] 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.

[42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[44] 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.

[45] J. Talukdar, S. Gupta, P. S. Rajpura, and R. S. Hegde, "Transfer learning for object detection using state-of-the-art deep neural networks," in *2018 5th international conference on signal processing and integrated networks (SPIN)*. IEEE, 2018, pp. 78–83.

[46] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.

[47] M. Li, J. Wu, X. Wang, C. Chen, J. Qin, X. Xiao, R. Wang, M. Zheng, and X. Pan, "Aligndet: Aligning pre-training and fine-tuning in object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6866–6876.

[48] F. J. P. Montalbo *et al.*, "A computer-aided diagnosis of brain tumors using a fine-tuned yolo-based model with transfer learning." *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 12, pp. 4816–4834, 2020.

[49] Y. Chen, X. Zhu, X. Gong, X. Yi, and S. Li, "Data poisoning attacks in internet-of-vehicle networks: Taxonomy, state-of-the-art, and future directions," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 20–28, 2022.

[50] Z. Xiong, H. Xu, W. Li, and Z. Cai, "Multi-source adversarial sample attack on autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2822–2835, 2021.

[51] H. Wang, C. Xiao, J. Kossaifi, Z. Yu, A. Anandkumar, and Z. Wang, "Augmax: Adversarial composition of random augmentations for robust training," *Advances in neural information processing systems*, vol. 34, pp. 237–250, 2021.

[52] C. Sun, Y. Cui, M. Ning, Y. Lu, D. Cao, and A. Khajepour, "Extending operational design domain for perception systems through robust learning," *IEEE Transactions on Intelligent Vehicles*, 2024.

[53] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[54] C.-C. Hsu, Y.-H. Tsai, Y.-Y. Lin, and M.-H. Yang, "Every pixel matters: Center-aware feature alignment for domain adaptive object detector," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer, 2020, pp. 733–748.

[55] C. Chen, Z. Zheng, X. Ding, Y. Huang, and Q. Dou, "Harmonizing transferability and discriminability for adapting object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8869–8878.

[56] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 749–757.

[57] Y.-J. Li, X. Dai, C.-Y. Ma, Y.-C. Liu, K. Chen, B. Wu, Z. He, K. Kitani, and P. Vajda, "Cross-domain adaptive teacher for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7581–7590.

[58] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready, "A robust learning approach to domain adaptive object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 480–490.

[59] J. Kronenberger and A. Haselhoff, "Dependency decomposition and a reject option for explainable models," *arXiv preprint arXiv:2012.06523*, 2020.

[60] L. Hacker and J. Seewig, "Insufficiency-driven dnn error detection in the context of sotif on traffic sign recognition use case," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 58–70, 2023.

[61] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.

[62] M. Keser, G. Schwalbe, A. Nowzad, and A. Knoll, "Interpretable model-agnostic plausibility verification for 2d object detectors using domain-invariant concept bottleneck models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3891–3900.

[63] G. Chen, Y. Zhang, and X. Li, "Attention-based highway safety planner for autonomous driving via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, 2023.

[64] C. Di Franco and N. Bezzo, "Interpretable run-time monitoring and replanning for safe autonomous systems operations," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2427–2434, 2020.

[65] C. Gall and N. Bezzo, "Gaussian process-based interpretable runtime adaptation for safe autonomous systems operations in unstructured environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 123–129.

[66] L. M. Schmidt, G. Kontes, A. Plinge, and C. Mutschler, "Can you trust your autonomous car? interpretable and verifiably safe reinforcement learning," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 171–178.

[67] W. Bao, Q. Yu, and Y. Kong, "Drive: Deep reinforced accident anticipation with visual explanation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7619–7628.

[68] L. H. Gilpin, V. Penubarthi, and L. Kagal, "Explaining multimodal errors in autonomous vehicles," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, pp. 1–10.

[69] A. Carlevaro and M. Mongelli, "A new svdd approach to reliable and explainable ai," *IEEE Intelligent Systems*, vol. 37, no. 2, pp. 55–68, 2021.

[70] A. Carlevaro, T. Alamo, F. Dabbene, and M. Mongelli, "Conformal predictions for probabilistically robust scalable machine learning classification," *Machine Learning*, vol. 113, no. 9, pp. 6645–6661, 2024.

[71] A. Radford, J. W. Kim, C. Hallacy *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[72] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*. PMLR, 2021, pp. 4904–4916.

[73] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[74] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[75] OpenAI, "Gpt-4v(ision) technical report," 2023, https://openai.com/research/gpt-4v-system-card.

[76] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: an embodied multimodal language model," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[77] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[78] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," in *European conference on computer vision*. Springer, 2024, pp. 256–274.

[79] T. Choudhary, V. Dewangan, S. Chandhok, S. Priyadarshan, A. Jain, A. K. Singh, S. Srivastava, K. M. Jatavallabhula, and K. M. Krishna, "Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 345–16 352.

[80] Y. Xu, Y. Hu, Z. Zhang, G. P. Meyer, S. K. Mustikovela, S. Srinivasa, E. M. Wolff, and X. Huang, "Vlm-ad: End-to-end autonomous driving through vision-language model supervision," *arXiv preprint arXiv:2412.14446*, 2024.

[81] H. Tian, K. Reddy, Y. Feng, M. Quddus, Y. Demiris, and P. Angeloudis, "Large (vision) language models for autonomous vehicles: Current trends and future directions," *Authorea Preprints*, 2024.

[82] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 46, no. 8, pp. 5625–5644, 2024.

[83] F.-L. Chen, D.-Z. Zhang, M.-L. Han, X.-Y. Chen, J. Shi, S. Xu, and B. Xu, "Vlp: A survey on vision-language pre-training," *Machine Intelligence Research*, vol. 20, no. 1, pp. 38–56, 2023.

[84] M. Naseer, S. Khan, and F. Porikli, "Indoor scene understanding in 2.5/3d for autonomous agents: A survey," *IEEE access*, vol. 7, pp. 1859–1887, 2018.

[85] J. Ni, Y. Chen, G. Tang, J. Shi, W. Cao, and P. Shi, "Deep learning-based scene understanding for autonomous robots: A survey," *Intelligence & Robotics*, vol. 3, no. 3, pp. 374–401, 2023.

[86] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "e-snli: Natural language inference with natural language explanations," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[87] A. Carlevaro, M. Lenatti, A. Paglialonga, and M. Mongelli, "Multiclass counterfactual explanations using support vector data description," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 6, pp. 3046–3056, 2023.

[88] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[89] A. Harakeh, M. Smart, and S. L. Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 87–93.

[90] L. Peng, H. Wang, and J. Li, "Uncertainty evaluation of object detection algorithms for autonomous vehicles," *Automotive Innovation*, vol. 4, no. 3, pp. 241–252, 2021.

[91] D. Miller, F. Dayoub, M. Milford, and N. Sünderhauf, "Evaluating merging strategies for sampling-based uncertainty techniques in object detection," in *2019 international conference on robotics and automation (icra)*. IEEE, 2019, pp. 2348–2354.

[92] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[93] D. Miller, N. Sünderhauf, H. Zhang, D. Hall, and F. Dayoub, "Benchmarking sampling-based probabilistic object detectors." in *CVPR Workshops*, vol. 3, 2019, p. 6.

[94] Z. Wang, Y. Li, Y. Guo, L. Fang, and S. Wang, "Data-uncertainty guided multi-phase learning for semi-supervised object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4568–4577.

[95] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[96] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1321–1330. [Online]. Available: https://proceedings.mlr.press/v70/guo17a.html

[97] M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, and M. Lucic, "Revisiting the calibration of modern neural networks," *Advances in neural information processing systems*, vol. 34, pp. 15 682–15 694, 2021.

[98] A. Rahimi, T. Mensink, K. Gupta, T. Ajanthan, C. Sminchisescu, and R. Hartley, "Post-hoc calibration of neural networks by g-layers," *arXiv preprint arXiv:2006.12807*, 2020.

[99] F. De Grancey, J.-L. Adam, L. Alecu, S. Gerchinovitz, F. Mamalet, and D. Vigouroux, "Object detection with probabilistic guarantees: A conformal prediction approach," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2022, pp. 316–329.

[100] S. Li, S. Park, X. Ji, I. Lee, and O. Bastani, "Towards pac multi-object detection and tracking," *arXiv preprint arXiv:2204.07482*, 2022.

[101] S. Park, O. Bastani, N. Matni, and I. Lee, "Pac confidence sets for deep neural networks via calibrated prediction," *arXiv preprint arXiv:2001.00106*, 2019.

[102] 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.

[103] A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster, "Conformal risk control," *arXiv preprint arXiv:2208.02814*, 2022.

[104] Y. Zhang, D. Zeng, J. Luo, X. Fu, G. Chen, Z. Xu, and I. King, "A survey of trustworthy federated learning: Issues, solutions, and challenges," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 6, Oct. 2024. [Online]. Available: https://doi.org/10.1145/3678181

[105] P. Cassará, A. Gotta, and L. Valerio, "Federated feature selection for cyber-physical systems of systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9937–9950, 2022.

[106] A. Li, H. Peng, L. Zhang, J. Huang, Q. Guo, H. Yu, and Y. Liu, "Fedsdg-fs: Efficient and secure feature selection for vertical federated learning," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[107] X. Li, R. Dowsley, and M. De Cock, "Privacy-preserving feature selection with secure multiparty computation," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 6326–6336. [Online]. Available: https://proceedings.mlr.press/v139/li21e.html

[108] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Fine-grained data selection for improved energy efficiency of federated edge learning," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3258–3271, 2022.

[109] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[110] J. Shin, Y. Li, Y. Liu, and S.-J. Lee, "Fedbalancer: data and pace control for efficient federated learning on heterogeneous clients," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 436–449. [Online]. Available: https://doi.org/10.1145/3498361.3538917

[111] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong, "Fedcorr: Multi-stage federated learning for label noise correction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 184–10 193.

[112] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural*

Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

[113] H. Cao, Q. Pan, Y. Zhu, and J. Liu, "Birds of a feather help: Context-aware client selection for federated learning," in *Proc. Int. Workshop Trustable Verifiable Auditable Federated Learn. Conjunction AAAI*, 2022, pp. 1–8.

[114] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," 2020. [Online]. Available: https://arxiv.org/abs/2010.01243

[115] Y. Jee Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., vol. 151. PMLR, 28–30 Mar 2022, pp. 10 351–10 375. [Online]. Available: https://proceedings.mlr.press/v151/jee-cho22a.html

[116] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 811–21 819, 2023.

[117] L. WANG, W. WANG, and B. LI, "Cmfl: Mitigating communication overhead for federated learning," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 954–964.

[118] J. L. Corcuera Bárcena, P. Ducange, A. Ercolani, F. Marcelloni, and A. Renda, "An approach to federated learning of explainable fuzzy regression models," in *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2022, pp. 1–8.

[119] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.

[120] A. Wilbik and P. Grefen, "Towards a federated fuzzy learning system," in *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2021, pp. 1–6.

[121] X. Zhu, D. Wang, W. Pedrycz, and Z. Li, "Horizontal federated learning of takagi–sugeno fuzzy rule-based models," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3537–3547, 2022.

[122] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn, M. Purcell, A. Rawat, T. Minh, N. Holohan, S. Chakraborty, S. Whitherspoon, D. Steuer, L. Wynter, H. Hassan, S. Laguna, M. Yurochkin, M. Agarwal, E. Chuba, and A. Abay, "Ibm federated learning: an enterprise framework white paper v0.1," 2020. [Online]. Available: https://arxiv.org/abs/2007.10987

[123] X. Chen, S. Zhou, B. Guan, K. Yang, H. Fao, H. Wang, and Y. Wang, "Fed-eini: An efficient and interpretable inference framework for decision tree ensembles in vertical federated learning," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 1242–1248.

[124] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi, "Privacy preserving vertical federated learning for tree-based models," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, p. 2090–2103, Jul. 2020. [Online]. Available: http://dx.doi.org/10.14778/3407790.3407811

[125] T. Dong, S. Li, H. Qiu, and J. Lu, "An interpretable federated learning-based network intrusion detection framework," 2022. [Online]. Available: https://arxiv.org/abs/2201.03134

[126] M. Polato, R. Esposito, and M. Aldinucci, "Boosting the federation: Cross-silo federated learning without gradient descent," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–10.

[127] R. Haffar, F. Naretto, D. Sánchez, A. Monreale, and J. Domingo-Ferrer, "Glor-flex: Local to global rule-based explanations for federated learning," in *Proceedings of the 2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Yokohama, Japan: IEEE, 2024, pp. 1–9.

[128] M. Paulik, M. Seigel, H. Mason, D. Telaar, J. Kluivers, R. van Dalen, C. W. Lau, L. Carlson, F. Granqvist, C. Vandevelde, S. Agarwal, J. Freudiger, A. Byde, A. Bhowmick, G. Kapoor, S. Beaumont, Áine Cahill, D. Hughes, O. Javidbakht, F. Dong, R. Rishi, and S. Hung, "Federated evaluation and tuning for on-device

personalization: System design & applications," 2021. [Online]. Available: https://arxiv.org/abs/2102.08503

[129] B. Soltani, Y. Zhou, V. Haghighi, and J. C. S. Lui, "A survey of federated evaluation in federated learning," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, E. Elkind, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2023, pp. 6769–6777, survey Track. [Online]. Available: https://doi.org/10.24963/ijcai.2023/758

[130] Z. Fan, H. Fang, X. Wang, Z. Zhou, J. Pei, M. P. Friedlander, and Y. Zhang, "Fair and efficient contribution valuation for vertical federated learning," 2025. [Online]. Available: https://arxiv.org/abs/2201.02658

[131] J. Fiosina *et al.*, "Explainable federated learning for taxi travel time prediction." in *VEHITS*, 2021, pp. 670–677.

[132] D. Janzing, L. Minorics, and P. Bloebaum, "Feature relevance quantification in explainable ai: A causal problem," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 2907–2916. [Online]. Available: https://proceedings.mlr.press/v108/janzing20a.html

[133] Z. Liu, Y. Chen, H. Yu, Y. Liu, and L. Cui, "Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, May 2022. [Online]. Available: https://doi.org/10.1145/3501811

[134] L. Nagalapatti and R. Narayanam, "Game of gradients: Mitigating irrelevant clients in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 9046–9054.

[135] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2597–2604.

[136] Q. Li, X. Wei, H. Lin, Y. Liu, T. Chen, and X. Ma, "Inspecting the running process of horizontal federated learning via visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4085–4100, 2022.

[137] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "Fate: An industrial grade platform for collaborative learning with data protection," *Journal of Machine Learning Research*, vol. 22, no. 226, pp. 1–6, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-815.html

[138] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu, Z. Zhang, W. Li, A. Myronenko, D. Yang, S. Yang, N. Rieke, A. Quraini, C. Chen, D. Xu, N. Ma, P. Dogra, M. Flores, and A. Feng, "Nvidia flare: Federated learning from simulation to real-world," 2022. [Online]. Available: https://arxiv.org/abs/2210.13291

[139] M. Ungersböck, T. Hiessl, D. Schall, and F. Michahelles, "Explainable federated learning: A lifecycle dashboard for industrial settings," *IEEE Pervasive Computing*, vol. 22, no. 1, pp. 19–28, 2023.

[140] P. Espinace, T. Kollar, A. Soto, and N. Roy, "Indoor scene recognition through object detection," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 1406–1413.

[141] O. Mattausch, D. Panozzo, C. Mura, O. Sorkine-Hornung, and R. Pajarola, "Object detection and classification from large-scale cluttered indoor scans," in *Computer Graphics Forum*, vol. 33, no. 2. Wiley Online Library, 2014, pp. 11–21.

[142] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*. Springer, 2024, pp. 529–545.

[143] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[144] M. Yaseen, "What is yolov8: An in-depth exploration of the internal features of the next-generation object detector. arxiv 2024," *arXiv preprint arXiv:2408.15857*, 2024.

[145] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *arXiv preprint arXiv:1702.07836*, 2017.

[146] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[147] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[148] G. Jocher *et al.*, "Yolov5: You only look once," https://github.com/ultralytics/yolov5, 2020, accessed: 2025-02-17.

[149] G. AI, "Open images dataset," 2023, available at https://storage.googleapis.com/openimages/web/visualizer/index.html.

[150] ——, "Open images dataset – wheelchair class," 2023, available at https://storage.googleapis.com/openimages/web/visualizer/index.html?type=detection&set=train&c=%2Fm%2F0qmmr.

[151] ——, "Open images dataset – bicycle class," 2023, available at https://storage.googleapis.com/openimages/web/visualizer/index.html?type=detection&set=train&c=%2Fm%2F0199g.

[152] ——, "Open images dataset – chair class," 2023, available at https://storage.googleapis.com/openimages/web/visualizer/index.html?type=detection&set=train&c=%2Fm%2F01mzpv.

[153] Roboflow, "Wheelchair detection dataset," 2023, available at https://universe.roboflow.com/2458761304-qq-com/wheelchair-detection/browse.

[154] L. Wang, J. Shi, G. Song, and I.-F. Shen, "Object detection combining recognition and segmentation," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2007, penn-Fudan Database for Pedestrian Detection and Segmentation.

[155] Y. Ma, C. Sun, J. Chen, D. Cao, and L. Xiong, "Verification and validation methods for decision-making and planning of automated vehicles: A review," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 480–498, 2022.

[156] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[157] S. Khastgir, S. A. Birrell, G. Dhadyalla, and P. A. Jennings, "The science of testing: An automotive perspective," *SAE Technical Papers*, vol. 2018, 2018.

[158] R. Palin, D. Ward, I. Habli, and R. Rivett, "Iso 26262 safety cases: Compliance and assurance," 2011.

[159] I. Iso, "Pas 21448-road vehicles-safety of the intended functionality," *International Organization for Standardization*, 2019.

[160] G. Li, Z. Ji, and X. Qu, "Stepwise domain adaptation (sda) for object detection in autonomous vehicles using an adaptive centernet," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 729–17 743, 2022.

[161] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *SAE Standard J3016_202104*, April 2021.

[162] A. Kuznietsov, B. Gyevnar, C. Wang, S. Peters, and S. V. Albrecht, "Explainable ai for safe and trustworthy autonomous driving: A systematic review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 12, pp. 19 342–19 364, 2024.

[163] C. Molnar, *Interpretable Machine Learning*. Christoph Molnar, 2023, https://christophm.github.io/interpretable-ml-book/.

[164] G. Schwalbe and B. Finzel, "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts," *Data Mining and Knowledge Discovery*, vol. 38, no. 5, pp. 3043–3101, 2024.

[165] Y. Peng, "Athec: A python library for computational aesthetic analysis of visual media in social science research," *Computational Communication Research*, vol. 4, no. 1, 2022.

[166] Y. Peng and J. B. JEMMOTT III, "Feast for the eyes: Effects of food perceptions and computer vision features on food photo popularity." *International Journal of Communication (19328036)*, vol. 12, 2018.

[167] J. Gu, Z. Han, S. Chen, A. Beirami, B. He, G. Zhang, R. Liao, Y. Qin, V. Tresp, and P. Torr, "A systematic survey of prompt engineering on vision-language foundation models," *arXiv preprint arXiv:2307.12980*, 2023.

[168] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, "The KTH-IDOL2 Database," KTH Royal Institute of Technology, CVAP/CAS, Stockholm,

Sweden, Tech. Rep. CVAP304, Oct. 2006. [Online]. Available: http://www.pronobis.pro/publications/luo2006idol2

[169] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[170] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world*. Springer, 2005.

[171] G. Shafer and V. Vovk, "A tutorial on conformal prediction." *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[172] A. M. Al-Khouri, "Data ownership: who owns "my data"," *International Journal of Management & Information Technology*, vol. 2, no. 1, pp. 1–8, 2012.

[173] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[174] K. Bonawitz, P. Kairouz, B. Mcmahan, and D. Ramage, "Federated learning and privacy," *Commun. ACM*, vol. 65, no. 4, p. 90–97, Mar. 2022. [Online]. Available: https://doi.org/10.1145/3500240

[175] R. Geetha and T. Thilagam, "A review on the effectiveness of machine learning and deep learning algorithms for cyber security," *Archives of Computational Methods in Engineering*, vol. 2, no. 4, pp. 2861–2879, 2021.

[176] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive iot networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4641–4654, 2020.

[177] D. Liu, L. Bai, T. Yu, and A. Zhang, "Towards method of horizontal federated learning: A survey," in *2022 8th International Conference on Big Data and Information Analytics (BigDIA)*. Qingdao, China: IEEE, 2022, pp. 259–266.

[178] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning: Concepts, advances, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3615–3634, 2024.

[179] G. Sartor and F. Lagioia, *The Impact of the General Data Protection Regulation (GDPR) on Artificial Intelligence*. Brussels, Belgium: Publications Office of the European Union, 2020.

[180] S. Huang, Y. Liang, F. Shen, and F. Gao, "Research on federated learning's contribution to trustworthy and responsible artificial intelligence," in *Proceedings of the 2024 3rd International Symposium on Robotics, Artificial Intelligence and Information Engineering*, ser. RAIIE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 125–129. [Online]. Available: https://doi.org/10.1145/3689299.3689322

[181] A. Samandari, M. Marchese, A. Paglialonga, F. Patrone, V. Rampa, and M. Mongelli, "Federated learning for rule-based systems: Preliminary studies," in *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*, IEEE. Lecco, Italy: IEEE, 2024, pp. 143–148.

[182] V. G. Costa and C. E. Pedreira, "Recent advances in decision trees: An updated survey," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4765–4800, 2023.

[183] S. Kianoush, S. Savazzi, V. Rampa, L. Costa, and D. Tolochenko, "A random forest approach to body motion detection: Multisensory fusion and edge processing," *IEEE Sensors Journal*, vol. 23, no. 4, pp. 3801–3814, 2023.

[184] D. Steinberg, "Cart: Classification and regression trees," in *The Top Ten Algorithms in Data Mining*. Boca Raton, FL: Chapman and Hall/CRC, 2009, pp. 193–216.

[185] J. Guo, P. Pietzuch, A. Paverd, and K. Vaswani, "Trustworthy ai using confidential federated learning," *Commun. ACM*, vol. 67, no. 9, p. 48–53, Aug. 2024. [Online]. Available: https://doi.org/10.1145/3677390

[186] M. Mongelli, M. Muselli, and E. Ferrari, "Achieving zero collision probability in vehicle platooning under cyber attacks via machine learning," in *2019 4th International Conference on System Reliability and Safety (ICSRS)*. Rome, Italy: IEEE, 2019, pp. 41–45.

[187] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887.

[188] A. M. Bouayed, A. Iaccovelli, and D. Naccache, "Sampling from autoencoders' latent space via quantization and probability mass function concepts," in *2023 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2023, pp. 1–10.

[189] M. Mongelli, E. Ferrari, M. Muselli, and A. Fermi, "Performance validation of vehicle platooning through intelligible analytics," *IET Cyber-Physical Systems: Theory & Applications*, vol. 4, no. 2, pp. 120–127, 2019.

[190] P. Pop, D. Scholle, I. Šljivo, H. Hansson, G. Widforss, and M. Rosqvist, "Safe cooperating cyber-physical systems using wireless communication: The safecop approach," *Microprocessors and microsystems*, vol. 53, pp. 42–50, 2017.

[191] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[192] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018. [Online]. Available: https://doi.org/10.21105/joss.00861

[193] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi, "Uniform manifold approximation and projection (umap)," in *Elements of Dimensionality Reduction and Manifold Learning*. Cham, Switzerland: Springer, 2023, pp. 479–497.

[194] B. Becker and R. Kohavi, "Adult," *UCI Machine Learning Repository*, vol. 10, p. C5XW20, 1996.

[195] A. Samandari, "Data: Federated learning for rule-based systems," [Online], accessed: Jul. 11, 2024, 2024, available: https://github.com/AhmadSamandari/Federated-Learning-for-Rule-Based-Systems.

[196] T. Ha, T. K. Dang, T. T. Dang, T. A. Truong, and M. T. Nguyen, "Differential privacy in deep learning: an overview," in *2019 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 2019, pp. 97–102.

[197] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 746–789, 2020.

New Approaches of eXplainable AI: From Video Analytics to Federated Learning

Ahmad Samandari