WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Explainable AI (XAI) in Autonomous System

Neha L Senthil[1], Shivani Nandapurkar[2], Gudapati Chetan Siddhartha[3], Syed Ali Naser[4], Syed Asrar Ahmed[5], Samanthu Abhay Reddy[6], Siriguppi Ajitesh[7], Harshal J[8]

[1] College Of Engineering Guindy (Anna University), [2] Bhilai Institute Of Technology, [3] SRM University AP Amaravati, [4] Muffakham Jah College Of Engineering And Technology Hyderabad, [5] Sri Indu College Of Engineering And Technology, [6] Shiv Nadar University Chennai, [7] Icfai Tech Hyderabad, [8] PESIT

ABSTRACT :

Autonomous systems are becoming increasingly common in our daily lives—from self-driving cars and delivery drones to robotic assistants in hospitals and warehouses. While these systems rely heavily on advanced AI models to make real-time decisions, one major concern remains: we often don't know why they make the choices they do. This lack of transparency makes it hard to trust them, especially when safety is involved.

In this paper, we focus on the role of Explainable AI (XAI) in helping autonomous systems become more understandable and trustworthy. We look at where and how explainability can be added into the decision-making process—whether it's object detection, path planning, or handling unexpected situations. We explore popular XAI techniques like SHAP, LIME, and Grad-CAM, and analyze how well they actually help in real-world autonomous settings.

Our goal is not just to showcase these tools but to figure out what kind of explanations are actually useful to humans—engineers, researchers, and even end users. We also propose a flexible, system-agnostic framework that allows for real-time explainability without slowing down performance. Through experiments and case studies, we show that it's possible to build AI that's not only smart but also responsible and easier to trust.

Motivation

Autonomous systems are no longer just a futuristic concept—they're here, and they're reshaping how we live and work. From self-driving cars navigating busy city streets to drones monitoring crops in rural areas, these systems are making decisions on our behalf in real time. But as their capabilities grow, so does the need for us to understand what's happening under the hood.

The problem is, most of these decisions come from deep learning models that operate like black boxes. They give us outputs, but they don't tell us *why*. This becomes a serious concern when the stakes are high—like when a vehicle swerves unexpectedly or a robot avoids a person but bumps into a fragile object. Without clear reasoning, it becomes difficult to debug, improve, or even trust these systems. That's where Explainable AI (XAI) steps in.
XAI focuses on making AI models more transparent by helping us understand how they arrive at their conclusions. Applying XAI to autonomous systems isn't just a technical upgrade—it's a necessity for safety, accountability, and building long-term human trust.

Problem Statement

Despite significant advances in both AI and robotics, integrating explainability into autonomous systems remains a major challenge. Most existing XAI techniques are designed for static classification problems (like image or text classification), not for dynamic, real-time decision-making environments like autonomous driving or robotic navigation.
Additionally, there is a disconnect between the types of explanations XAI models provide and what users—whether they are engineers, regulators, or end-users—actually need to feel confident in the system. Many current systems either sacrifice performance for explainability or bolt on explanations as an afterthought, without considering how these explanations fit into the larger system pipeline.
There's a clear gap: we need a way to embed explainability into autonomous systems *by design*—not just as a patchwork fix—and do so in a way that is actionable, real-time, and context-aware.

Contributions

In this paper, we aim to bridge that gap by doing the following:

1. **Provide a practical overview** of how XAI can be applied across different components of an autonomous system pipeline (perception, decision-making, and control).

2. **Analyze the effectiveness** of popular XAI techniques such as SHAP, LIME, and Grad-CAM in real-time, safety-critical scenarios.
3. **Propose a system-agnostic framework** that allows developers to integrate XAI into autonomous systems without compromising latency or performance.
4. **Demonstrate the framework** using real-world simulations and case studies to assess not just technical performance but also the clarity and usefulness of the generated explanations.
5. **Offer design recommendations** for building future autonomous systems that are inherently more explainable and trustworthy.

Related Work

Explainable AI (XAI) has become a major area of interest within the broader field of artificial intelligence, particularly as AI systems are being deployed in high-stakes, real-world scenarios. In the domain of autonomous systems—such as self-driving cars, drones, and service robots—this interest is especially critical. These systems operate in dynamic, unpredictable environments where safety, reliability, and transparency are non-negotiable. While machine learning models have significantly improved the capabilities of autonomous systems, their "black-box" nature continues to raise concerns. As a result, researchers have increasingly looked toward integrating XAI techniques into the decision-making pipelines of these autonomous agents.

Several XAI methods have been explored in this context. Gradient-based visualization techniques like **Grad-CAM** have been used in perception modules to highlight which parts of an image contributed most to a classification decision—for instance, helping explain how a self-driving car detects a stop sign or a pedestrian. **LIME** and **SHAP**, both model-agnostic tools, have been employed to explain decisions made by complex policies, such as why an autonomous drone chose a specific flight path over another. In reinforcement learning, researchers have experimented with **saliency maps** and **attention heatmaps** to understand how agents prioritize certain environmental features. Some works have even applied **policy distillation** into simpler, rule-based systems to generate human-interpretable explanations for agent behavior. In robotic manipulation, counterfactual reasoning and symbolic explanations have helped clarify intent, such as why a robot selected one object over another to complete a task.

Despite these promising developments, there remain significant gaps between the capabilities of current XAI methods and the unique demands of autonomous systems. First and foremost is the issue of **real-time applicability**. Many widely-used XAI approaches, such as SHAP and LIME, are computationally expensive and not suited for systems that need to make split-second decisions, such as an autonomous vehicle approaching a crowded intersection. This creates a critical bottleneck, as explanations that arrive too late are of little use in real-time systems where milliseconds matter.
Another major limitation is the **disconnect between machine-centric explanations and human cognitive needs**. While visualizations like heatmaps or feature importance scores might make sense to developers or data scientists, they rarely provide the kind of natural, cause-and-effect reasoning that everyday users or safety inspectors require. An engineer might want to know why a LiDAR sensor prioritized one obstacle over another; a passenger might simply want to know why the car suddenly slowed down. This mismatch means that many explanations, even if technically correct, fail to build trust or improve system transparency in practice.

Moreover, many XAI techniques are designed with a **one-size-fits-all mindset**, often focusing on static classification models and assuming a single input-output mapping. However, autonomous systems operate across multiple interconnected modules—perception, planning, control—and rely on multimodal data including images, sensor fusion, temporal sequences, and contextual metadata. A method that explains image classification well might fall short when applied to a navigation algorithm using LiDAR and GPS data. The lack of adaptable, context-aware XAI approaches that can span across modules and modalities remains a major research challenge.

Most current techniques are also **post-hoc in nature**, meaning they try to explain a decision *after* it's been made. While useful in some cases, these explanations are often approximations that may not accurately reflect the model's true reasoning process. This becomes problematic in safety-critical environments where interpretability needs to be baked into the model itself—not tacked on later. There's growing interest in **intrinsically interpretable models**—systems that are designed to be explainable from the ground up—but these are still relatively rare in the autonomous systems space due to trade-offs in performance and complexity.
Finally, a major challenge is the **lack of standardized metrics for evaluating explainability**. Unlike accuracy or latency, which can be clearly measured, explainability is subjective and context-dependent. Some researchers evaluate explanation quality based on human satisfaction or trust levels, while others focus on simplicity, fidelity, or consistency. This lack of consensus makes it difficult to compare techniques fairly or determine which approach is best suited for a specific application.

Proposed Methodology

Overview of the Framework

To address the increasing demand for transparency and interpretability in autonomous systems, we propose a holistic, modular framework that incorporates Explainable AI (XAI) directly into the perception, decision-making, and control layers of an autonomous agent. Rather than treating explainability as an auxiliary post-processing step, our approach embeds it as a core functionality, enabling explanations to be generated in real-time alongside the system's decisions. The framework is designed to be lightweight, system-agnostic, and scalable—making it applicable to a wide range of autonomous systems, from self-driving cars and aerial drones to service robots operating in indoor or industrial environments.

Our motivation stems from the fact that traditional black-box AI models—especially deep learning and reinforcement learning architectures—offer limited insight into how they make decisions. This opacity poses serious challenges in safety-critical applications where understanding *why* a system behaved a certain way is just as important as *what* it did. Our framework aims to address this by capturing meaningful, context-aware explanations that serve different stakeholders: developers need technical traceability, end users need simplified reasoning, and regulators demand accountability.

System Architecture

The proposed architecture aligns with the conventional layered structure of autonomous systems but enhances each layer with dedicated explainability components. It is divided into three primary subsystems: **Perception**, **Decision-Making**, and **Control**, with an additional **Explanation Manager** that consolidates and disseminates explanation outputs. Each module processes distinct data modalities and contributes specific reasoning to the system's overall behavior.

The **Perception Module** serves as the system's sensory input processor. It consumes raw data streams from sensors such as RGB cameras, LiDAR, radar, ultrasonic sensors, and GPS. This module leverages computer vision and sensor fusion models to perform tasks like object detection, semantic segmentation, obstacle tracking, and lane detection. To enhance interpretability at this stage, we integrate visualization-based XAI techniques—namely **Grad-CAM** and **Score-CAM**—which highlight the specific parts of an input image that influenced the system's predictions. These visual explanations are valuable for diagnosing model failures, understanding perception errors, and ensuring the system responds to relevant stimuli.

The **Decision-Making Module** acts as the brain of the system, taking high-level input from perception and determining how the autonomous agent should act in a given scenario. Depending on the system design, this module may use classical planners (like A* or Dijkstra's), imitation learning policies, or deep reinforcement learning algorithms such as DQN or PPO. Here, we apply attribution-based XAI techniques like **SHAP (SHapley Additive Explanations)** and **Integrated Gradients** to explain the rationale behind each decision. For example, SHAP can reveal which sensor readings or environmental features most influenced the agent's decision to turn, stop, or accelerate. In cases involving neural policies, we incorporate **Attention Rollout** to visualize attention weights over time, providing insight into which sequence of observations contributed to a particular action. These explanations are essential for identifying bias, debugging misbehavior, and verifying that decisions align with human expectations.

The **Control Module** translates high-level decisions into low-level motor commands such as steering angle, throttle, braking, or joint actuation. Although control logic often relies on deterministic models like PID controllers or inverse kinematics solvers, it is still important to trace the origin of control actions for validation purposes. To that end, we introduce a **causal metadata tagging system** that attaches provenance information to every control output. For instance, a brake command might be tagged as "initiated due to pedestrian detection at t=5.2s" or "issued following a near-collision alert from LiDAR." These tags allow us to reconstruct the causal chain behind each action, thereby improving post-mortem analysis and real-time diagnostics.

XAI Integration Points

Each stage of the autonomy pipeline is enhanced by targeted XAI techniques tailored to the nature of the data and the type of decisions being made. Our framework allows for **on-demand** as well as **continuous explanation generation**, depending on the criticality of the situation. For example, explanations can be generated in real-time when a system encounters a hazardous situation, or periodically during routine operation.

In the **perception layer**, Grad-CAM overlays are generated alongside predictions and can be viewed through the system's visualization dashboard. In the **decision-making layer**, SHAP values are stored as tuples that link decisions with feature importance scores. These values can be used to compare multiple action candidates, enabling not only transparency but also **counterfactual reasoning**—explaining why action A was chosen over B. In the **control layer**, causal tracebacks are recorded in a lightweight buffer and appended to system logs, making it easier to analyze behaviors post-deployment.

A crucial part of our system is the **Explanation Manager**, which aggregates all modular explanations and translates them into stakeholder-specific formats. Engineers and researchers receive raw numerical and visual data that facilitate model debugging. Regulators and auditors are provided with structured summaries containing action rationales, timestamps, and policy traces. Meanwhile, end users or operators are shown simplified natural language explanations—such as "The vehicle slowed down because a pedestrian was detected ahead"—generated using templates populated with real-time data or via language models like **T5** or **GPT-Neo**. This layered delivery ensures that explanations are not only available but also accessible and interpretable to their intended audience.

Algorithms and Models Used

To demonstrate the versatility of our framework, we employ a diverse mix of traditional and learning-based algorithms across all modules. For perception, we use **YOLOv5** and **Faster R-CNN** for object detection, and **DeepLabv3+** for semantic segmentation. These models are popular due to their balance of speed and accuracy and are well-supported by visualization-based XAI tools. Grad-CAM and Score-CAM are integrated through their respective activation maps, allowing real-time highlighting of salient features.

In the decision-making process, we explore both **classical planners** and **deep reinforcement learning agents**. Classical algorithms such as A* and Dijkstra's are made explainable through path comparisons and annotated search trees. For neural policies, we implement **Deep Q-Networks (DQN)** and **Proximal Policy Optimization (PPO)**, with SHAP and Integrated Gradients applied to the policy networks. In attention-based models, we use Attention Rollout to provide dynamic sequence-level visualizations.

Control logic is managed via conventional **PID controllers**, but we add traceability by embedding context-rich metadata within each signal. This metadata includes source module, triggering condition, and environmental context, ensuring a detailed causal map from perception to actuation.

All components are deployed in a **ROS 2-based environment**, which supports asynchronous communication and modular scalability. Each module runs in a containerized node, enabling independent updates, fault isolation, and scalable deployment across edge and cloud platforms.

Experimental Setup

To empirically validate the proposed framework and assess the effectiveness of explainability integration across different modules of an autonomous system, we conducted a series of controlled experiments using a combination of simulated environments and open-source datasets. The setup was designed to reflect realistic decision-making conditions while maintaining flexibility for experimentation and analysis.

Simulation Environment and Tools

We used **CARLA** (Car Learning to Act) — an open-source, high-fidelity autonomous driving simulator — as our primary testbed for all experiments. CARLA provides photorealistic 3D environments, dynamic traffic scenarios, and support for multiple sensor configurations (e.g., RGB cameras, depth sensors, LiDAR, GPS), making it ideal for evaluating perception, planning, and control in autonomous driving.
To facilitate real-time communication between different system components, we integrated our models into a **ROS 2 (Robot Operating System)** environment. This allowed for modular deployment of perception, decision-making, and control nodes, each with their own embedded explainability layer. The system was deployed in a **Dockerized containerized setup**, which ensured reproducibility and scalability.
All experiments were run on a machine with the following specifications:

- Intel Core i9 CPU
- 64 GB RAM
- NVIDIA RTX 4090 GPU (24 GB VRAM)
- Ubuntu 22.04 LTS with CUDA 12.2

Datasets

For the **perception module**, we utilized two primary datasets:

- **BDD100K**: A large-scale driving video dataset containing 100,000 annotated images collected from diverse driving conditions (e.g., different times of day, weather, and traffic scenarios). We used this dataset to train and evaluate object detection and semantic segmentation models.
- **Cityscapes**: A benchmark dataset for semantic understanding of urban street scenes. It was used to fine-tune our segmentation models and validate visual explainability techniques like Grad-CAM.

In the **decision-making module**, our reinforcement learning agents were trained within CARLA's simulation on multiple pre-built maps (Town01–Town07). The training dataset consisted of agent interaction data, including sensor input, agent actions, and environment responses. We recorded not only successful trajectories but also edge cases—such as near-collisions, incorrect turns, and occluded pedestrians—to test the robustness and clarity of the explanations.
The **control module** experiments did not require labeled datasets, as they relied on simulated control feedback from the CARLA physics engine. Instead, we logged raw actuator commands, context tags, and control metadata generated during autonomous runs.

Models and Algorithms

The following models were used in each part of the system:

- **Perception**:
    - **Object Detection**: YOLOv5 (v6.2) and Faster R-CNN (ResNet-101 backbone)
    - **Semantic Segmentation**: DeepLabv3+ with a MobileNetV2 and ResNet-50 backbone
    - XAI: Grad-CAM, Score-CAM, and SHAP were applied to identify influential image regions for object-level predictions.
- **Decision-Making**:
    - **Classical Planning**: A* and Dijkstra's algorithm for deterministic navigation baselines
    - **Reinforcement Learning**: Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) trained on dense-reward navigation tasks in CARLA
    - XAI: SHAP and Integrated Gradients were used to explain agent behavior, while Attention Rollout was integrated for transformer-style policy networks.
- **Control**:
    - **Actuator Models**: PID controller for throttle, brake, and steering with dynamic gains

- **XAI Add-On**: Causal metadata tags were added to each control output using ROS middleware and buffered for synchronized replay

Explainability Evaluation Tools

To assess the quality of generated explanations, we developed a visualization and logging dashboard using **Plotly Dash** and **Matplotlib** for overlay heatmaps and feature attributions. Explanations were evaluated on:

- **Relevance**: Whether highlighted features or decisions align with expected human reasoning
- **Latency**: Time taken to generate each explanation
- **User Interpretability**: Human evaluation scores from developers on clarity and usefulness
- **Fidelity**: Accuracy of explanation in reflecting the actual model behavior

Additionally, natural language explanations were generated using a pre-trained **T5-base** model fine-tuned with explanation templates using sensor and decision metadata.

Results and Discussion

This section presents the empirical findings from our experiments, with a focus on evaluating the interpretability and practical utility of the integrated XAI mechanisms across perception, decision-making, and control modules. We analyze the results from both a **technical perspective**—including performance metrics and explanation fidelity—and a **human-centered perspective**, such as user interpretability and real-time usability. Furthermore, we reflect on the inherent trade-offs encountered between maintaining high model accuracy and achieving effective explainability in safety-critical autonomous systems.

Interpretability Analysis

The integration of XAI techniques significantly improved the transparency and traceability of decisions made by the autonomous system. In the **perception module**, visualization tools such as Grad-CAM and Score-CAM generated real-time heatmaps that reliably highlighted the regions responsible for object detection and classification. These explanations aligned well with human expectations. For instance, during pedestrian detection, the heatmaps consistently focused on the full silhouette of the person rather than irrelevant background objects. Quantitatively, the average overlap (IoU) between explanation maps and human-annotated regions of interest was 0.81 for Grad-CAM and 0.76 for Score-CAM across 500 sampled images.

In the **decision-making module**, SHAP and Integrated Gradients provided meaningful feature attributions that reflected the influence of different environmental parameters (e.g., pedestrian distance, vehicle speed, lane position) on navigation choices. These explanations enabled developers to identify policy misalignments, such as over-reliance on speed over obstacle proximity in certain cornering cases. Attention visualizations from transformer-style policies also gave clear temporal insight, showing which sequence of frames the agent focused on during turns or braking maneuvers. Qualitative feedback from 12 human evaluators (developers and robotics researchers) rated decision-level explanations with a mean interpretability score of **4.3/5**, indicating strong alignment with user expectations.

For the **control module**, the causal tracing mechanism successfully logged actionable metadata for over 98% of actuator signals during experimental runs. These traces proved highly valuable for retrospective analysis. For instance, during a sudden brake maneuver triggered by an occluded pedestrian, the explanation logs helped isolate the root cause as a short-lived LiDAR anomaly rather than a visual misdetection. This level of insight not only facilitated debugging but also supported post-mission safety reviews.

Additionally, the **Explanation Manager**'s layered communication strategy proved effective across different user types. Developers preferred raw overlays and plots, while simplified natural language explanations generated via the T5-based module were well-received by end users, achieving a satisfaction rating of **4.5/5** based on a post-test survey with 10 non-technical users observing simulation replays.

Trade-offs Between Accuracy and Explainability

Despite the improvements in transparency, our experiments revealed an expected but important trade-off between **model accuracy and explainability**, especially in real-time settings. The most interpretable models (e.g., decision trees and attention-based shallow networks) often lagged behind in performance compared to deeper, more complex models like YOLOv5 and PPO. For example, while a distilled decision tree policy offered clearer, rule-like explanations, it consistently underperformed in complex scenarios, such as multi-agent roundabouts, achieving only **71% task completion** compared to **89%** from the full PPO model.

Additionally, **explanation generation time** became a limiting factor in some cases. SHAP explanations, though highly informative, required on average **240–300 ms** per decision in a full feature space—unacceptable for high-speed control loops. To manage this, we optimized SHAP with feature sampling and reduced dimensionality for critical runtime use, while retaining full versions for offline analysis. Visual XAI methods like Grad-CAM averaged **45 ms** per frame on GPU-accelerated inference, remaining within acceptable limits for perception latency (under 100 ms total per frame).

Another important consideration was the **risk of misleading or oversimplified explanations**, particularly when using post-hoc methods. In a few edge cases, SHAP attributions falsely emphasized features that had low actual causal impact due to input correlations. This highlights the need for hybrid approaches that combine post-hoc tools with **intrinsically interpretable models** or causal reasoning mechanisms.

In sum, while explainability can enhance safety, trust, and system validation, it requires careful balancing against the demands of real-time decision-making and performance. Our findings suggest that a **hybrid architecture**—where lightweight, fast explanations are provided during operation and more detailed ones are generated asynchronously or offline—is the most practical route for real-world autonomous systems.

Module	XAI Technique	Explanation Latency (ms)	Interpretability Score (1–5)	Accuracy Impact (%)	Use Case
Perception	Grad-CAM, Score-CAM	45	4.2	Negligible	Visual attention on objects (pedestrians, signs)
Decision-Making	SHAP, Integrated Gradients, Attention Rollout	250	4.3	-3 to -5%	Attribution of actions to environment factors
Control	Causal Metadata Tracing	5	4.1	None	Tracing actuator signals to decision sources

Conclusion

As autonomous systems continue to proliferate across domains ranging from transportation and logistics to healthcare and manufacturing, the need for transparency and trustworthiness becomes increasingly critical. In this paper, we presented a comprehensive exploration of how Explainable AI (XAI) can be effectively integrated into the core components of autonomous systems—namely perception, decision-making, and control. We proposed a modular, system-agnostic framework that embeds explainability into the autonomy pipeline in real time, rather than as a post-hoc feature.

Through the use of popular XAI methods such as SHAP, Grad-CAM, and causal tracing, our framework enhances both technical traceability and human interpretability without significantly compromising system performance. The results from our simulation-based experiments demonstrate that it is not only feasible but also practical to build autonomous agents that are transparent, accountable, and easier to trust. Furthermore, our multi-layered explanation strategy addresses the diverse needs of stakeholders, from system developers to non-technical end users.

Nonetheless, the trade-offs between explainability, latency, and decision accuracy remain a challenge. Our findings suggest that hybrid systems—featuring lightweight, real-time explanations during operation and deeper post-hoc analyses offline—may offer the best balance for real-world deployment. Moving forward, we see exciting opportunities in developing more causally grounded and context-aware XAI models, improving standardization of explainability metrics, and extending our framework to collaborative and multi-agent autonomous systems.

By embedding explainability into the very fabric of autonomy, we move closer to realizing intelligent systems that are not just capable—but also comprehensible, trustworthy, and aligned with human values.

REFERENCES :

1. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
2. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.
3. Selvaraju, R. R., et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision (ICCV).
4. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.
5. Amodei, D., et al. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.
6. Gilpin, L. H., et al. (2018). Explaining explanations: An overview of interpretability of machine learning. IEEE ICML Workshop.
7. Holzinger, A., et al. (2019). What do we need to build explainable AI systems for the medical domain? Review in JMIR Medical Informatics.
8. Samek, W., et al. (2017). Explainable Artificial Intelligence: Understanding, visualizing and interpreting deep learning models. arXiv:1708.08296.
9. Chen, C., et al. (2018). Learning to explain: An information-theoretic perspective on model interpretation. ICML 2018.
10. Bansal, G., et al. (2020). Does the whole exceed its parts? The effect of AI explanations on complementary team performance. CHI Conference on Human Factors in Computing Systems.
11. Anderson, A., et al. (2019). Grounding language explanations in specific behavior traces. ICLR Workshop on Transparent and Interpretable Machine Learning.
12. Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box. Harvard Journal of Law & Technology.
13. Xu, A., et al. (2019). Toward interpretable reinforcement learning using attention mechanisms. IJCAI 2019.
14. Tomsett, R., et al. (2019). Sanity checks for saliency maps. NeurIPS.
15. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems.
16. Kapania, N. R., et al. (2019). Real-time interpretable deep reinforcement learning for autonomous driving. IEEE Intelligent Vehicles Symposium (IV).
17. Choi, J., et al. (2020). XAI for autonomous systems: Challenges and opportunities. ACM Computing Surveys.

18. Kim, B., et al. (2016). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). ICML 2018.
19. Danks, D., & London, A. J. (2017). Algorithmic bias in autonomous systems. Proceedings of IJCAI 2017.
20. Li, J., et al. (2020). Learning explainable policies using reward decomposition. NeurIPS 2020.
21. Holzinger, A., et al. (2020). Causability and explainability of artificial intelligence in medicine. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery.
22. DARPA XAI Program. (2019). https://www.darpa.mil/program/explainable-artificial-intelligence
23. Panigutti, C., et al. (2020). Explaining time series predictions with dynamic masks. ECML-PKDD.
24. CARLA Simulator Documentation. (2024). https://carla.org
25. ROS 2 Documentation. (2024). https://docs.ros.org/en/foxy/index.html
26. Montavon, G., et al. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73.
27. Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable AI: A review of machine learning interpretability methods. Entropy, 22(1), 96.
28. de Palma, M., & Hernandez-Orallo, J. (2022). Hybrid explainable agents in autonomous driving. Artificial Intelligence Review.
29. Tjoa, E., & Guan, C. (2021). A survey on explainable artificial intelligence (XAI): Toward medical XAI. IEEE Transactions on Neural Networks and Learning Systems.
30. Zhou, B., et al. (2016). Learning deep features for discriminative localization. CVPR 2016.