Review

Synchronization, Optimization, and Adaptation of Machine Learning Techniques for Computer Vision in Cyber-Physical Systems: A Comprehensive Analysis

Kai Hung Tang , Mohamed Chahine Ghanem * , Pawel Gasiorowski , Vassil Vassilev , Karim Ouazzane

Posted Date: 19 August 2025

doi: 10.20944/preprints202501.0521.v3

Keywords: machine learning; computer vision; cyber-physical systems; CPS; adaptation; synchronisation; optimisation

Review

Synchronisation, Optimisation, and Adaptation of Machine Learning Techniques for Computer Vision in Cyber-Physical Systems: A Comprehensive Analysis

Kai Hung Tang [1], Mohamed Chahine Ghanem [2,*], Pawel Gasiorowski [1], Vassil Vassilev [1] and Karim Ouazzane [1]

[1] Cyber Security Research Centre, London Metropolitan University, London, UK
[2] Cybersecurity Institute, Department of Computer Science, University of Liverpool, and Cyber Security Research Centre, London Metropolitan University, London, UK
* Correspondence: ghanemm@staff.londonmet.ac.uk

Abstract

Cyber-Physical Systems (CPS) seamlessly integrate computers, networks, and physical devices, enabling machines to communicate, process data, and respond to real-world conditions in real time. By bridging the digital and physical worlds, CPS ensures operations that are efficient, safe, innovative, and controllable. As smart cities and autonomous machines become more prevalent, understanding CPS is crucial for driving future progress. Recent advancements in edge computing, AI-driven vision, and collaborative systems have significantly enhanced CPS capabilities. synchronisation, optimisation, and adaptation are intricate processes that impact CPS performance across different domains. Therefore, identifying emerging trends and uncovering research gaps is essential to highlight areas that require further investigation and improvement. This systematic review and analysis aims to offer a unique point to researchers and facilitates this process by allowing researchers to benchmark and compare various techniques, evaluate their effectiveness, and establish best practices. It provides evidence-based insights into optimal strategies for implementation while addressing potential trade-offs in performance, resource usage, and reliability. Additionally, such reviews help identify widely accepted standards and frameworks, contributing to the development of standardised approaches.

Keywords: machine learning; computer vision; cyber-physical systems; CPS; adaptation; synchronisation; optimisation

1. Introduction

1.1. Context and Importance

This paper focuses on the integration of machine learning (ML) techniques with computer vision (CV) to address the evolving demands of cyber-physical systems (CPS). CPS, which combines computational and physical processes, is increasingly dependent on Computer Vision (CV) for real-time perception and decision-making. These systems span various applications, including autonomous vehicles, smart grids, industrial automation, healthcare devices, and intelligent transportation networks. The real-time capabilities provided by CV enable CPS to interpret complex visual data from their environment, facilitating tasks such as object detection, scene understanding, and adaptive control.

However, synchronising and optimising ML models for such applications remains a critical challenge, given CPS's dynamic and resource-constrained nature. Key issues include ensuring low-latency processing, maintaining accuracy under varying operational conditions, and efficiently managing computational resources, particularly in embedded or edge-computing scenarios. Furthermore, CPS often operates in unpredictable and sometimes harsh environments, requiring robust ML models that handle noisy or incomplete data without compromising performance.

Another dimension of the challenge involves the continuous adaptation of ML algorithms to evolving data patterns and system behaviours. CPS needs adaptive learning strategies to update models in real-time or near-real-time. This demands advanced techniques such as incremental learning, transfer learning, and federated learning, which allow models to evolve based on new information without the need for complete retraining from scratch.

This paper explores these multifaceted challenges, reviewing recent advancements and identifying key areas for future research. By addressing these issues, we aim to pave the way for more efficient, reliable, and adaptable ML-integrated CV solutions in next-generation CPSs.

1.2. Problem Statement

Despite advancements in ML and CV, their deployment in CPS faces several challenges:

1. Synchronisation issues due to heterogeneous hardware and real-time constraints. CPS environments often consist of diverse hardware components. Ensuring seamless integration and real-time data processing across these heterogeneous platforms is complex. Synchronisation becomes particularly challenging when multiple sensors and processing units work together to provide a coherent and timely response. Variations in processing power, data transfer rates, and latency can lead to discrepancies or delays undermining the system's overall performance. Addressing these issues requires sophisticated algorithms and synchronisation protocols that can harmonise the operation of different hardware components while meeting stringent real-time constraints.
2. Optimisation difficulties related to balancing accuracy and computational efficiency. ML models, particularly deep learning architectures, often demand substantial computational resources to achieve high accuracy. In CPS, where real-time decision-making is crucial, striking a balance between model performance and computational efficiency is essential. Resource-constrained environments, such as embedded systems or edge devices, may not have the capacity to run large models or handle intensive computations. Therefore, optimising models to deliver accurate predictions without overloading system resources is a significant challenge. Techniques such as model pruning, quantisation, and knowledge distillation are commonly explored, but implementing them effectively without compromising performance remains an ongoing area of research.
3. Adaptation requirements to ensure robust performance across varying environments and tasks. CPS often operates in dynamic and unpredictable environments where conditions can change rapidly. For instance, an autonomous vehicle must adapt to different weather conditions, lighting variations, and traffic scenarios. Similarly, industrial CPS must handle fluctuations in sensor data and operational conditions. ML models trained in controlled settings may struggle to maintain accuracy when faced with such variability. This necessitates adaptive learning strategies and robust models capable of generalising across different tasks and environments. Techniques such as transfer learning, online learning, and domain adaptation are crucial, but integrating them into CPS without causing disruptions or requiring constant retraining poses significant challenges.

Addressing these challenges is essential for the effective deployment of ML for CV in CPS, ensuring these systems can operate reliably, efficiently, and safely in real-world applications. This paper explores potential solutions and innovations aimed at overcoming these hurdles, paving the way for more resilient and adaptable CPS architectures.

1.3. Objectives

We aim to synthesize existing research on ML techniques for CV in CPS. This involves examining a wide range of methodologies, including traditional approaches, advanced deep learning architectures, and other methods, to understand their applications, strengths, and limitations. The review will cover various CV tasks relevant to CPS, such as object detection, image classification, semantic segmentation, and anomaly detection. By analysing existing literature, we intend to highlight the most effective strategies, key milestones, and technological advancements that have shaped this interdisciplinary field. This synthesis will serve as a foundation for understanding how ML-driven CV solutions contribute to enhancing the functionality and reliability of CPS across different domains, including autonomous vehicles, smart manufacturing, and healthcare systems. Despite significant progress, this review seeks to identify and analyse existing gaps related to synchronisation, optimisation, and adaptation. In terms of synchronisation, we will examine the complexities of integrating heterogeneous hardware components and maintaining real-time performance across diverse CPS platforms. For optimisation, we will explore the trade-offs between computational efficiency and model accuracy, particularly in resource-constrained environments. Regarding adaptation, we aim to uncover the limitations of current ML models in handling dynamic and unpredictable environments, where robust performance is essential. By systematically identifying these gaps, we hope to provide a clearer picture of the unresolved issues that need to be addressed to enable more effective and reliable ML-CV integration in CPS.

This review will propose future directions for research and development in this interdisciplinary domain. The recommendations will focus on key areas such as developing more efficient and adaptable algorithms, enhancing real-time synchronisation frameworks, and designing robust models capable of operating under varying conditions. Additionally, we will highlight the importance of interdisciplinary collaboration and domain-specific experts in addressing these complex challenges holistically. Emerging trends, such as edge computing, federated learning, and hybrid models combining symbolic reasoning with neural networks, will also be discussed as potential avenues for innovation. By outlining these future directions, we aim to inspire further research and development efforts, ultimately contributing to the evolution of smarter, more efficient, and resilient CPSs.

1.4. Structure

The remainder of this article is organised as follows: Chapter 2 outlines the systematic review process. Chapter 3 provides foundational knowledge on ML, CV, and CPS. Chapter 4 synthesises key findings and identifies emerging themes. Chapter 5 evaluates current research and explores future opportunities. Finally, Chapter 6 offers practical insights and Chapter 7 summarises the significance of the study. The structure of this article is illustrated in Figure 1.

doi:10.20944/preprints202501.0521.v3

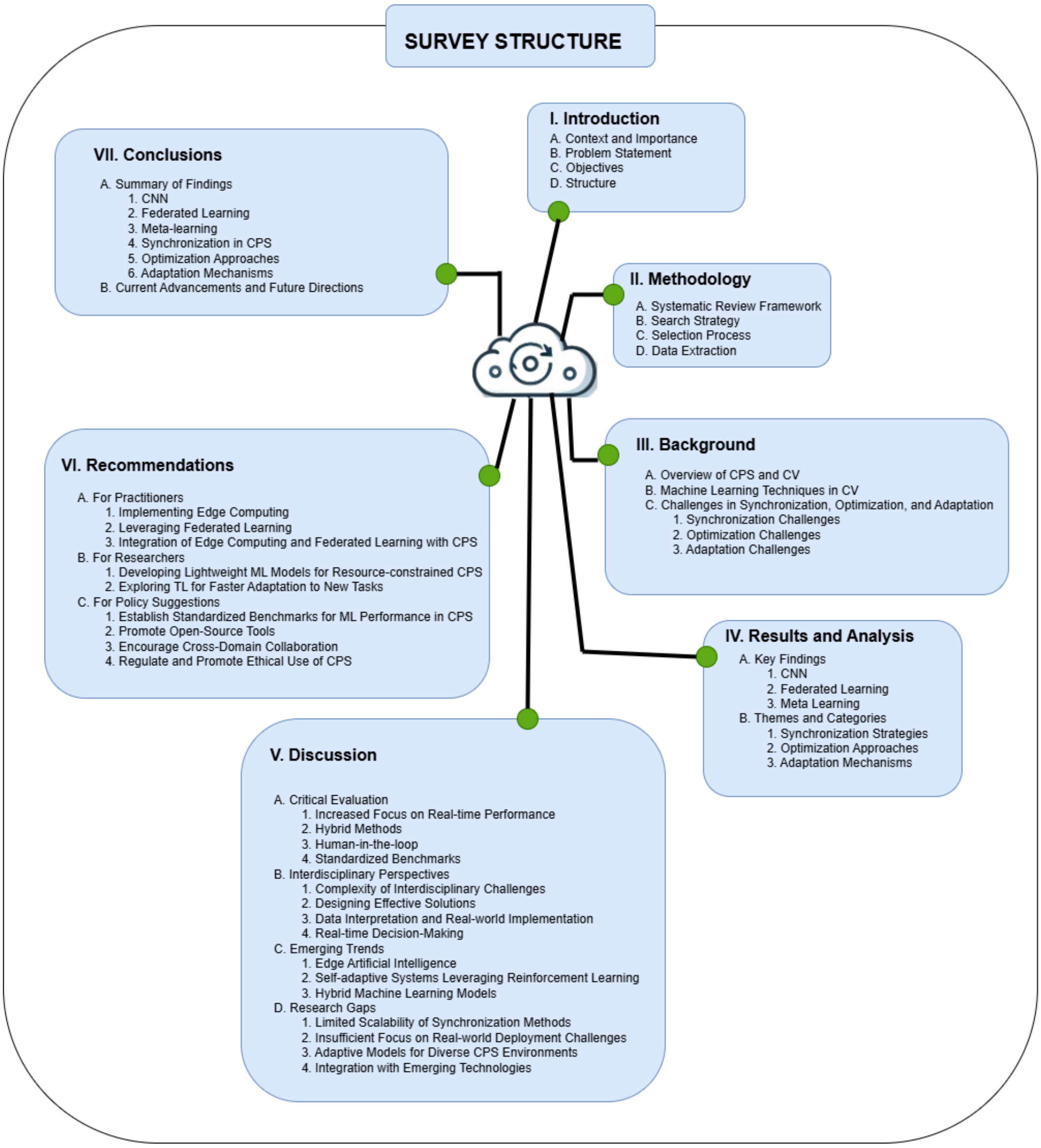
SURVEY STRUCTURE
I. Introduction
A. Context and Importance
B. Problem Statement
C. Objectives
D. Structure
II. Methodology
A. Systematic Review Framework
B. Search Strategy
C. Selection Process
D. Data Extraction
III. Background
A. Overview of CPS and CV
B. Machine Learning Techniques in CV
C. Challenges in Synchronization, Optimization, and Adaptation
1. Synchronization Challenges
2. Optimization Challenges
3. Adaptation Challenges
IV. Results and Analysis
A. Key Findings
1. CNN
2. Federated Learning
3. Meta Learning
B. Themes and Categories
1. Synchronization Strategies
2. Optimization Approaches
3. Adaptation Mechanisms
V. Discussion
A. Critical Evaluation
1. Increased Focus on Real-time Performance
2. Hybrid Methods
3. Human-in-the-loop
4. Standardized Benchmarks
B. Interdisciplinary Perspectives
1. Complexity of Interdisciplinary Challenges
2. Designing Effective Solutions
3. Data Interpretation and Real-world Implementation
4. Real-time Decision-Making
C. Emerging Trends
1. Edge Artificial Intelligence
2. Self-adaptive Systems Leveraging Reinforcement Learning
3. Hybrid Machine Learning Models
D. Research Gaps
1. Limited Scalability of Synchronization Methods
2. Insufficient Focus on Real-world Deployment Challenges
3. Adaptive Models for Diverse CPS Environments
4. Integration with Emerging Technologies
VI. Recommendations
A. For Practitioners
1. Implementing Edge Computing
2. Leveraging Federated Learning
3. Integration of Edge Computing and Federated Learning with CPS
B. For Researchers
1. Developing Lightweight ML Models for Resource-constrained CPS
2. Exploring TL for Faster Adaptation to New Tasks
C. For Policy Suggestions
1. Establish Standardized Benchmarks for ML Performance in CPS
2. Promote Open-Source Tools
3. Encourage Cross-Domain Collaboration
4. Regulate and Promote Ethical Use of CPS
VII. Conclusions
A. Summary of Findings
1. CNN
2. Federated Learning
3. Meta-learning
4. Synchronization in CPS
5. Optimization Approaches
6. Adaptation Mechanisms
B. Current Advancements and Future Directions

Figure 1. Structure of the Article.

2. Methodology

2.1. Systematic Review Framework

The review follows a systematic framework, adhering to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [1] guidelines, ensuring a thorough and transparent evaluation of the relevant literature. The PRISMA framework involves several critical steps, including developing a detailed research protocol, conducting comprehensive and reproducible literature searches across multiple databases, and applying predefined inclusion and exclusion criteria for study selection.

By adhering to these guidelines, the review minimises bias and enhances reliability. The methodology involves a two-phase screening process (title/abstract and full-text reviews) conducted by independent reviewers, and discrepancies resolved by consensus. Data extraction is performed using standardised forms to capture key study characteristics, findings, and quality assessments. In addition, a PRISMA flow diagram is presented to visually illustrate the search process, the number of studies identified, screened, and included, as well as the reasons for exclusions.

This systematic approach ensures comprehensive coverage of the literature and facilitates transparency and replicability, enabling other researchers to validate and build upon the findings.

2.2. Search Strategy

When conducting academic research, we have used multiple scholarly databases can ensure comprehensive coverage of relevant literature. Databases like IEEE Xplore, SpringerLink, Scopus, and Google Scholar provide unique advantages for finding peer-reviewed articles and conference papers.

For the search process, the following keywords have been used: Machine Learning, Computer Vision in Cyber-Physical Systems, synchronisation in Machine Learning, and optimisation and Adaptation of Computer Vision Algorithms. Start by entering keywords into title/abstract and then into full-text reviewers, we combine the keywords with AND or OR to explore related works and access citations.

To consider the most recent works in the field, the search period is limited between 2010 and 2024. However, in some cases, it was necessary to use older preliminary references to get an overview of all the basic notions and fully cover the study's topic. Only papers on ML for CV, emphasising studies addressing synchronisation, optimisation, or adaptation in CPS have been considered. Inclusion criteria focused on peer-reviewed publications from 2010 to 2023, emphasising studies addressing synchronisation, optimisation, or adaptation in CPS.

2.3. Selection Process

Initially, keywords were entered into the title and abstract search fields to identify articles directly addressing the core research topics. Following this preliminary screening, full-text reviews were conducted to assess the relevance and depth of the selected works concerning our research objectives. Boolean operators such as AND and OR were used to combine these keywords, allowing us to refine searches, link interconnected concepts, and identify relevant citations more effectively. By strategically utilising comprehensive databases and systematically enhancing search methodologies, we aimed to construct a robust overview of the current research landscape, highlighting existing gaps and opportunities for future exploration.

To assess the quality and relevance of the studies, we utilised established metrics such as citation impact and methodological rigour. Additionally, we assigned a qualitative score ranging from 0 to 5 to evaluate how effectively each study addressed our research questions. A score of 5 indicated a strong alignment between the study's research question and ours, without suggesting duplication. This scoring system provided a structured framework for systematically evaluating the relevance and comprehensiveness of each study within the context of our research objectives.

The temporal distribution of the selected articles, shown in Figure 2, reveals a notable upward trend, with a significant surge in publications over the last two years. This trend underscores the growing interest and rapid acceleration in research focusing on ML algorithms for CV applications within CPS.

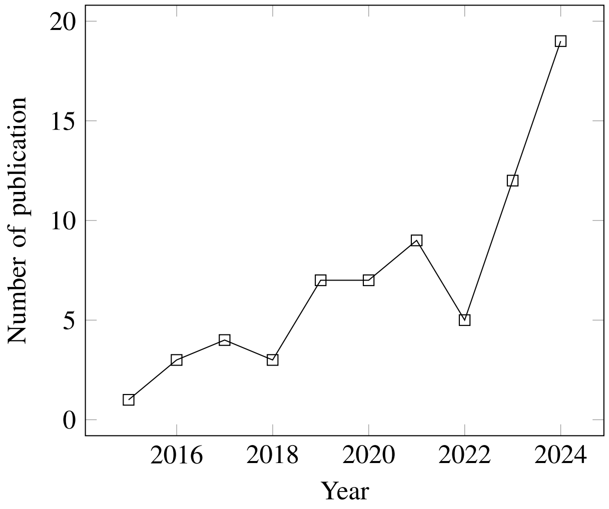

Figure 2. Distribution of the publications between 2015 and 2024.

doi:10.20944/preprints202501.0521.v3

2.4. Data Extraction

Data extraction involved identifying and recording key data points critical to our studies for CV applications in CPS. The first data category focused on ML models and architectures utilised, including specific algorithms, frameworks, and design patterns employed in the selected articles. This information was vital for understanding the underlying computational approaches and their suitability for CPS applications. Another important area of focus was the synchronisation strategies between ML algorithms and CPS hardware. This encompassed methods to ensure smooth integration and coordination between the computational components of ML systems and the physical processes controlled by CPS. Details included timing mechanisms, communication protocols, and any co-design considerations.

We also extracted information on optimisation techniques for resource-constrained environments, emphasising strategies used to adapt ML operations for hardware with limited computational power, energy, or memory. These data points provided insights into practical implementations where resource efficiency was a critical constraint.

Lastly, we gathered data on adaptation methods for dynamic operational contexts, which included techniques used to modify or retrain ML models in response to changing environmental conditions or system demands. This category highlighted how studies addressed the challenges of real-time adaptability and resilience in CPS applications. The search article flow chart is shown in Figure 3. The literature selection process is shown in Figure 4.

Collectively, these data points formed a comprehensive basis to analyse trends, innovations, and gaps in CV application to CPS, allowing for a robust evaluation of current methodologies and their implications.

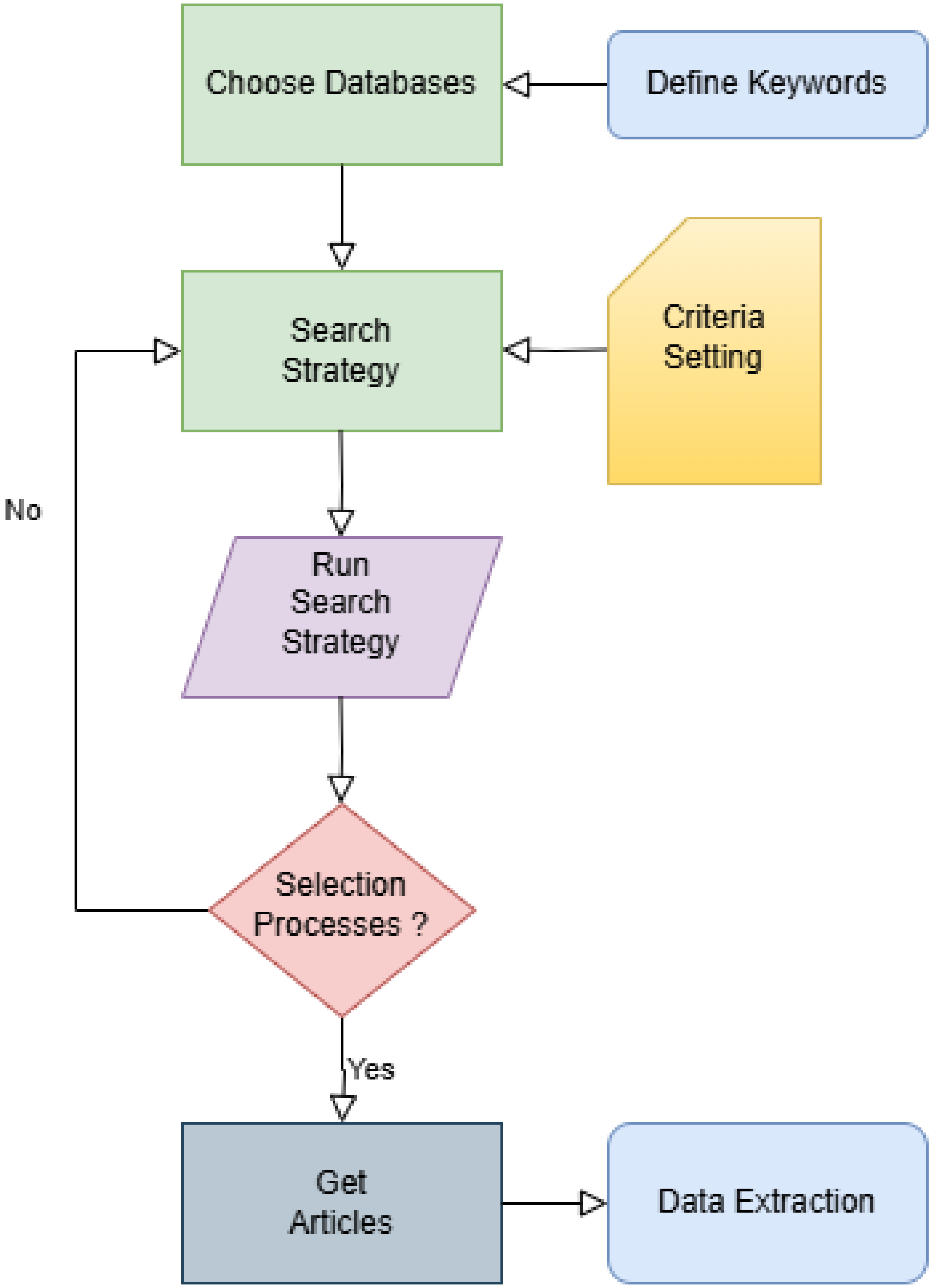

Figure 3. Papers Search and Review Flowchart.

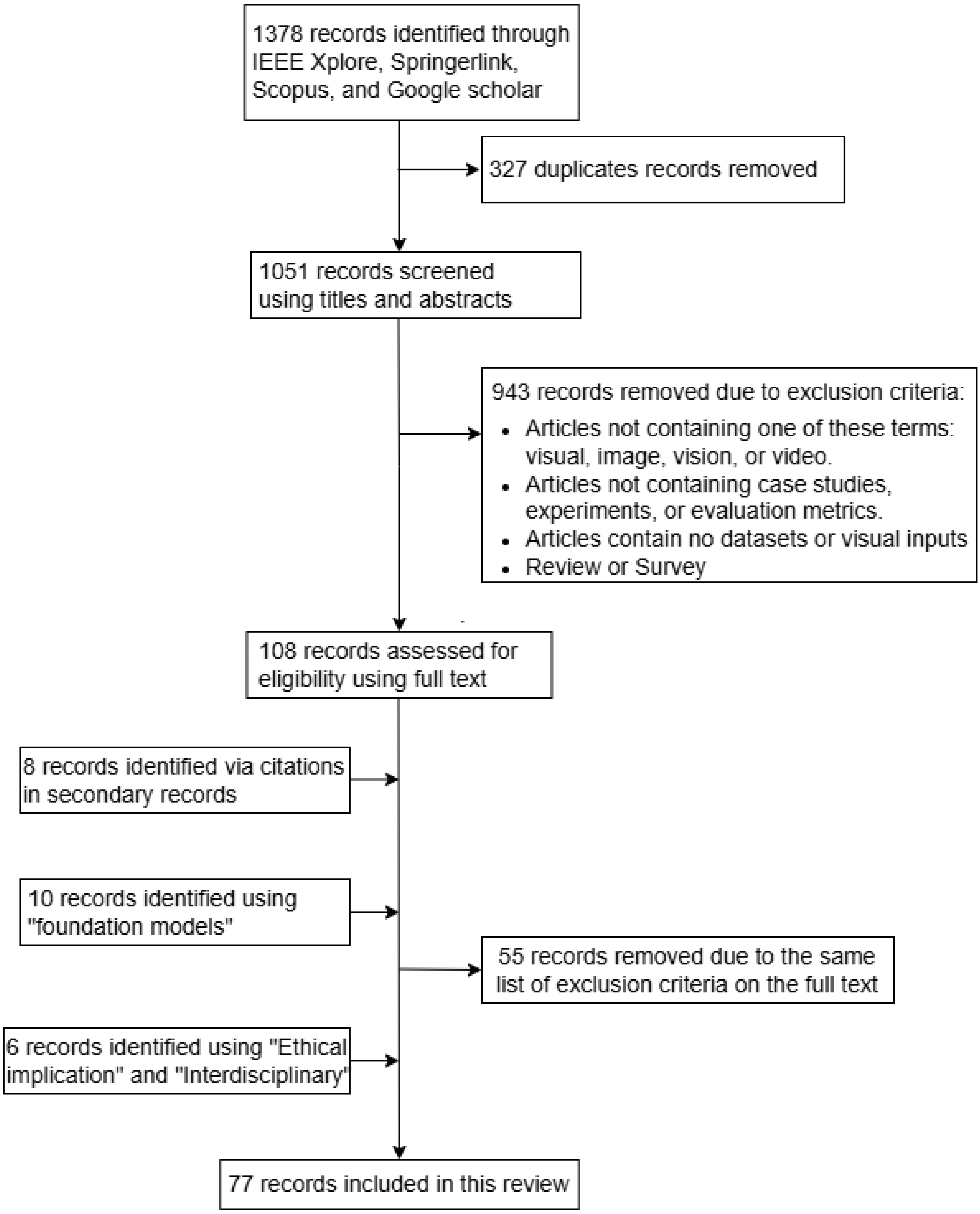
1378 records identified through IEEE Xplore, Springerlink, Scopus, and Google scholar
327 duplicates records removed
1051 records screened using titles and abstracts
943 records removed due to exclusion criteria:
• Articles not containing one of these terms: visual, image, vision, or video.
• Articles not containing case studies, experiments, or evaluation metrics.
• Articles contain no datasets or visual inputs
• Review or Survey
108 records assessed for eligibility using full text
8 records identified via citations in secondary records
10 records identified using "foundation models"
55 records removed due to the same list of exclusion criteria on the full text
6 records identified using "Ethical implication" and "Interdisciplinary"
77 records included in this review

Figure 4. PRISMA Flowchart Diagram of Literature Selection Process.

3. Background

3.1. Overview of Cyber-physical Systems (CPS) and Computer Vision (CV)

CPSs integrate computing elements with physical processes to enable real-time monitoring and control. These systems bridge the physical and digital worlds, driving advancements in smart grids, autonomous vehicles, industrial automation, and healthcare. Figure 5 illustrates CPS application domains.

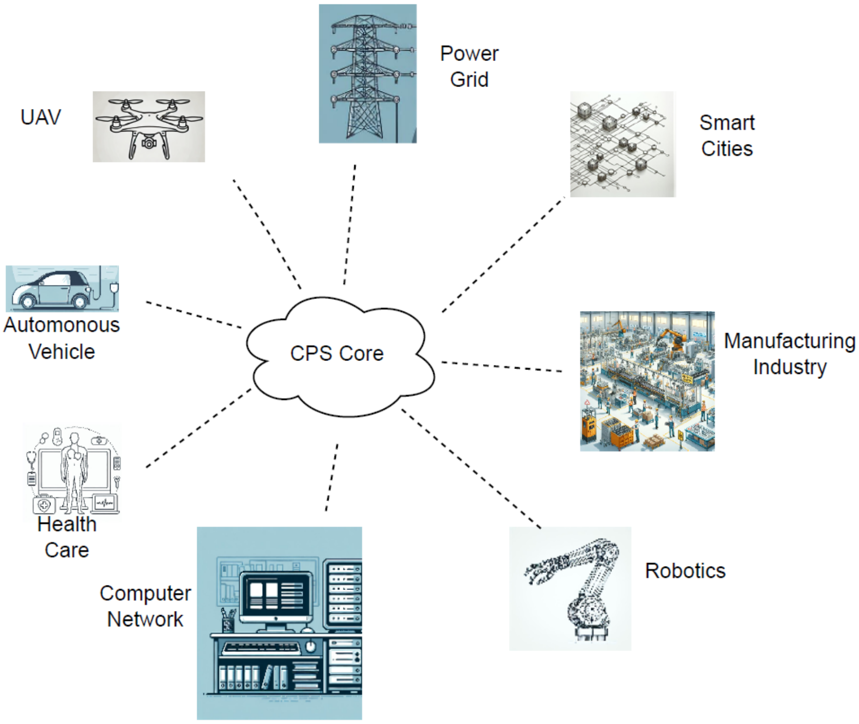

Figure 5. CPS Domaines and Applications.

The core components of CPS include:

- Sensors: collect data from the physical environment, converting real-world information into digital signals. Examples include temperature sensors, cameras, LIDAR, GPS, and accelerometers. In CPS, sensors play a vital role in:
  - Monitoring environmental conditions (e.g., in smart buildings).
  - Detecting anomalies in industrial processes.
  - Providing input for control decisions in autonomous vehicles.
- Actuators: perform actions based on decisions made by the computational units, transforming digital commands into physical actions. They can control various devices, such as motors, valves, or robotic arms. Key functions include:
  - Adjusting machinery operations in manufacturing.
  - Steering autonomous vehicles based on sensor data.
  - Regulating power distribution in smart grids.
- Computational Units: process sensor data, run control algorithms, and send commands to actuators. They can range from embedded microcontrollers to powerful cloud-based systems. Functions include:
  - Real-time data analysis.
  - Running predictive models to anticipate system behaviours.
  - Ensuring system security and reliability through robust software protocols.

Recent advancements in edge computing, AI-driven vision, and collaborative systems continue to extend CPS capabilities. The functioning of CPS is grounded in real-time data from the physical environment to guide decision-making and actions. CV enhances CPS in the ways below:

3.1.1. Perception and Sensing

CV acts as the "eyes" of CPS, gathering visual data through cameras and sensors. It is critical for autonomous vehicles, drones, and industrial robots, where vision algorithms extract features for object recognition, motion detection, depth estimation, and tracking to have real-time scene understanding.

An example application in autonomous vehicles is that CV detects pedestrians, vehicles, traffic signals, and road conditions to provide inputs for the control system.

3.1.2. Real-time Monitoring and Feedback

CPS relies on real-time feedback from the physical environment to function efficiently. Computer vision (CV) facilitates this by capturing and interpreting visual data in real-time, enabling systems to dynamically adjust their actions or decisions based on changes in their surroundings. One of the defining features of CPS is its real-time operation. In industrial environments, robots continuously monitor production, identify flaws, optimise performance, and anticipate potential issues to prevent malfunctions. This minimises human intervention while enhancing production speed. Similarly, CPS-enabled drones navigate and avoid obstacles during deliveries, while smart home systems automatically adjust lighting and temperature in response to current conditions.

3.1.3. Autonomy and Decision Making

CPSs harness CV, AI, and ML to enable autonomous decision-making. Vision systems analyse large volumes of visual data, identify patterns, and make independent decisions without human intervention. For example, drones use vision-based navigation to autonomously avoid obstacles and inspect critical infrastructure such as bridges and power lines. In healthcare, CPS supports continuous patient monitoring through wearable devices, robotic surgical tools, and advanced prosthetics, delivering accurate and timely medical care. Similarly, smart grids optimise energy efficiency by monitoring consumption and distributing power more effectively, reducing waste and improving overall resource management.

3.1.4. Safety and Surveillance

CV plays a vital role in safety and security, particularly in smart cities and industries. Vision-based systems can detect objects, identify faces or license plates, and trigger alarms in response to suspicious activity. Vision-enabled surveillance systems in smart grids monitor critical infrastructure for breaches or abnormalities caused by intrusions or equipment malfunctions. Self-driving cars also depend on CPS to process vast amounts of data in real time, enabling split-second decisions that enhance road safety.

3.1.5. Human-machine Interfaces

CV enables human-machine interfaces by interpreting human gestures, motions, or expressions, allowing systems to interact with humans in real time. It is widely used in smart gadgets, healthcare, and robotics. In healthcare, vision systems track patient movements or facial expressions to monitor health conditions or assist in physical therapy.

3.2. Machine Learning Techniques in CV

The following section explores commonly used ML models in CV, highlighting their architectures, functionalities, and applications:

3.2.1. Convolutional Neural Networks (CNNs) for Image Recognition

CNNs are cornerstones in computer vision, designed specifically to handle grid-like data such as images. Inspired by the human visual cortex, CNNs use a series of convolutional layers to automatically and adaptively learn spatial hierarchies of features from input images. These networks apply filters (kernels) that slide over the image, detecting patterns such as edges, textures, and complex objects at different layers. CNNs are widely used for tasks such as image classification, object detection, and semantic segmentation. A schematic diagram of a basic CNN architecture is shown in Figure 6 and key components of CNNs include:

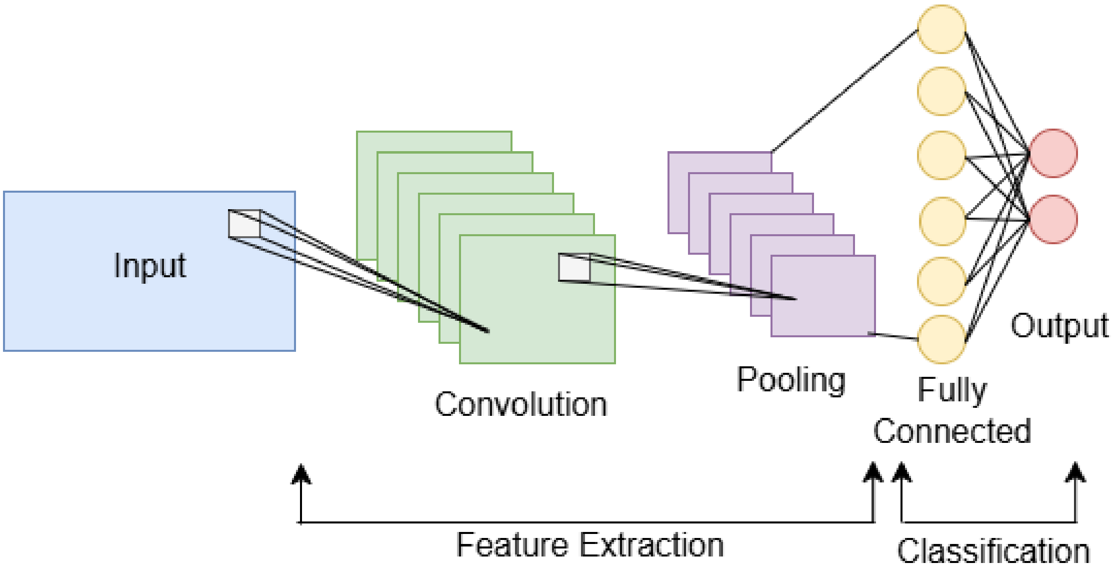

Figure 6. Schematic diagram of a basic CNN architecture [2].

- Convolutional Layers: These layers apply convolution operations to the input image, using filters (or kernels) to detect various features such as edges, textures, and patterns.
- Activation Functions: After convolution, activation functions are applied to introduce non-linearity, helping the network learn more complex patterns.
- Pooling Layers: These layers reduce the dimensionality of feature maps, preserving essential information while minimising computational load and making the network more robust to variations in input.
- Fully Connected Layers: After several convolutional and pooling layers, these layers combine the features to make predictions or classifications.
- Output Layer: The final layer usually uses a softmax activation function to produce a probability distribution over the possible classes, allowing the network to make a prediction.

3.2.2. Recurrent Neural Networks (RNNs) for Sequential Data Processing

While primarily designed for sequential or time-series data, RNNs have found applications in computer vision, particularly in tasks involving sequences of images or video data. RNNs are unique in their ability to process sequences by maintaining a hidden state that captures information about previous elements in the sequence. This makes them effective for modelling temporal dependencies, making them useful for tasks like stock price prediction and weathering forecasting. By analysing frames sequentially, RNNs can be used for tasks like action recognition in videos. For instance, they can process sequences of video frames to recognise activities (e.g., walking, running) or generate textual descriptions for images.

Figure 7 illustrates the variants of RNNs [3], accompanied by their descriptions below:

- Long Short-Term Memory (LSTM): is a type of RNN designed to address the vanishing gradient problem, allowing them to capture long-term dependencies more effectively.
- Gated Recurrent Unit (GRU): are a simplified version of LSTMs that also help mitigate the vanishing gradient problem while being computationally more efficient.

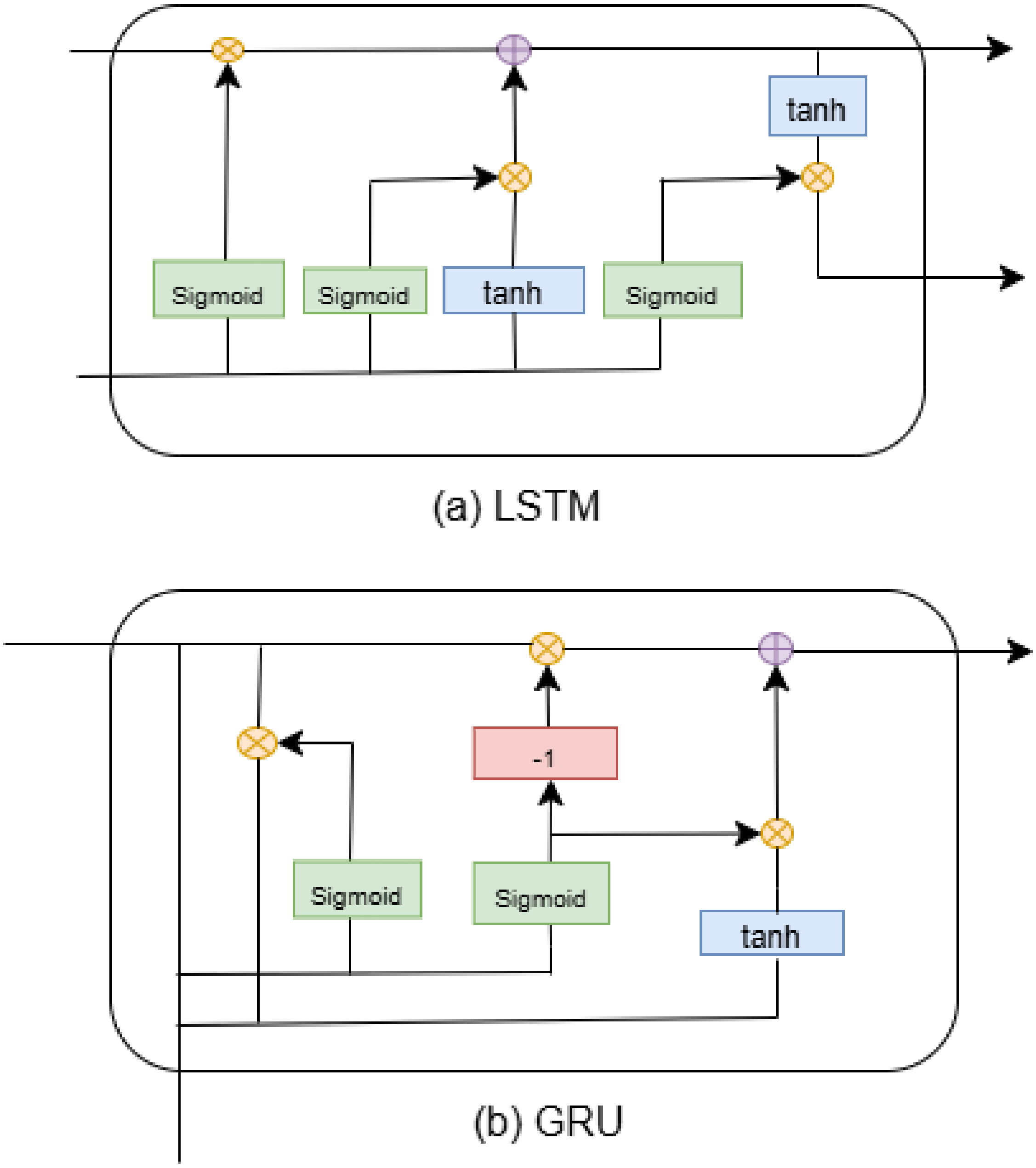

Figure 7. Basic structure of the a) LSTM and b) GRU neural networks. [3].

3.2.3. Transformer-Based Architectures for Advanced Feature Extraction

Transformers, initially developed for natural language processing (NLP), have transformed deep learning with their attention mechanisms, enabling models to capture global relationships within input sequences. In computer vision, architectures such as the Vision Transformer (ViT) apply these principles to image data, offering robust feature extraction and representation capabilities. A view of the model is shown in Figure 8 [4]. Transformer-based models perform exceptionally well in tasks like image classification, object detection, and segmentation. Their core concepts include the self-attention mechanism and patch embedding. The self-attention mechanism allows models to assess the importance of different image regions, capturing long-range dependencies and contextual relationships. Patch embedding converts an image into a sequence of fixed-size patches, analogous to word tokens in NLP.

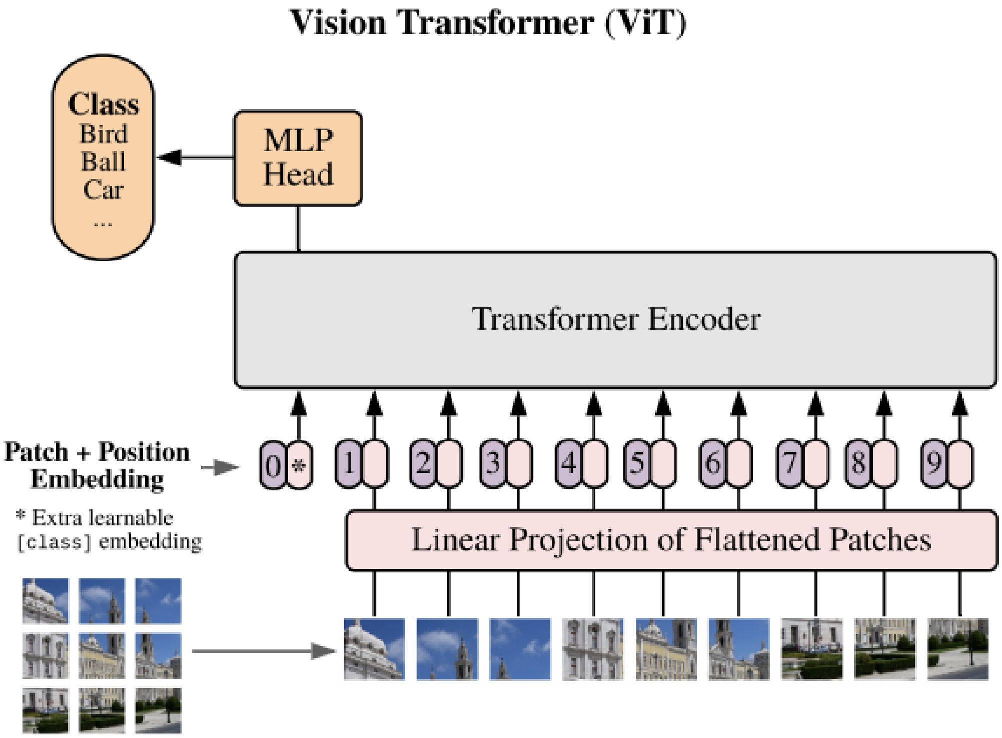
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 8. A Vision Transformer Model [4].

3.3. *Challenges in Synchronisation, Optimisation, and Adaptation*

Numerous applications [5], such as driverless cars, smart cities, healthcare monitoring, industrial automation, and robotics, become possible when CV and CPS are integrated. However, this integration has several significant challenges, especially when it comes to synchronising and optimising machine learning algorithms. These challenges are the following:

3.3.1. Synchronisation Challenges

CPS requires synchronisation since it involves several subsystems operating in real-time, frequently in dispersed contexts. The following difficulties arise while integrating a CV with CPS:

- Real-time Data Fusion: CV systems process visual data alongside other sensors like LiDAR, RADAR, and accelerometers. Poor decision-making may result from system lags or timestamp misalignments.
- Latency in Decision Making: The processing of deep learning-based CV algorithms is time-consuming, making real-time synchronisation with CPS controls essential. Delays can compromise safety in systems like autonomous vehicles and drones.
- Distributed Processing: Coordinating CV tasks among nodes in a distributed CPS network is challenging, particularly while handling time-sensitive communications and preserving system dependability.

3.3.2. Optimisation Challenges

Efficient CV algorithms are crucial for real-world CPS applications, but optimising them poses significant hurdles, including:

- Resource Constraints: Memory and processing power on CPS devices, particularly edge devices, are frequently constrained. Because deep learning models require many resources, optimising them within these limitations might be challenging.

- Model Efficiency: Techniques like model compression and pruning are necessary to reduce the size and complexity of neural networks for tasks such as object detection and recognition on resource-constrained edge devices.
- Real-time optimisation: There is a trade-off between time performance and accuracy, particularly challenging for low-latency applications like autonomous navigation.
- Communication Bandwidth: In distributed CPS, efficiently transmitting high-dimensional CV data requires methods such as video compression and local processing using edge computing.

3.3.3. Adaptation Challenges

CPS adopts flexible and adaptable CV algorithms under dynamic environments. Key challenges are the following:

- Dynamic Environments: CV algorithms must adapt continuously to changing conditions, such as variations in lighting, weather, and the presence of new obstacles, unlike static conditions.
- Transfer Learning and Domain Adaptation: It is challenging to adapt pre-trained models to new environments with minimal retraining, such as when autonomous vehicles move from urban to rural areas.
- Online Learning and Incremental Updates: CPS requires real-time model updates without requiring full retraining, which is computationally costly, due to continuous data streaming.
- Handling Uncertainty and Noise: To ensure accurate decision-making for managing noisy, incomplete, or uncertain sensor data, the method should be robust.

The relevant techniques and methods identified in the references to address these challenges are summarized in Table 1.

Table 1. Relevant works that attempted to address Synchronisation, Optimisation and Adaptation challenges.

Challenges	**Reference**	**Techniques/Methods**
Real-time Data Fusion	[6] [7] [8] [9] [10]	A hybrid framework integrating an FCNx and an EKF Known templates method, using predefined patterns; An information-theoretic approach. LFF YOLO Network The digital twin architecture integrates the different modules. Multi-sensor fusion algorithms
Latency in Decision Making	[8] [9] [11] [12]	LFF YOLO Network The publish-subscribe pattern architecture Parallel Algorithm for Multitarget Tracking Coordinate transformation or homography to map 2D face coordinates onto the 3D space
Distributed Processing	[6] [?]	A hybrid framework integrating an FCNx and an EKF A distributed motion control system for reconfigurable manufacturing systems
Resource Constraints	[13] [14]	Compressed MobileNet V3 Architecture Key optimisation techniques including distributed optimisation algorithms, gradient compression
Model Efficiency	[15] [10]	Pruning: Removes unimportant connections; Quantisation: Huffman Coding: weight sharing Dynamic power management; Efficient algorithms; Hardware optimisation
Real-time optimisation	[16] [17] [14]	A hyperheuristic multi-objective evolutionary search method CNN for mobile computer vision systems Federated learning and neuromorphic computing
Communication Bandwidth	[14] [10]	Communication-efficient algorithms, including Ring All Reduce and decentralized training methods Open communication protocols
Dynamic Environments	[18]	A physical-virtual interactive parallel light fields collection method
Transfer Learning and Domain Adaptation	[19] [20]	Neural style transfer and GAN ResADM: a transfer-learning-based attack detection method
Online Learning and Incremental Updates	[21]	Model-Agnostic Meta-Learning and Conditional Neural Processes
Handling Uncertainty and Noise	[22]	Ensemble model

4. Results and Analysis

4.1. Key Findings

4.1.1. CNN

CNN plays a dominant role in CV applications within CPS due to its specialized design for image analysis. Its layered architecture is highly effective at automatically learning patterns, features, and spatial hierarchies from images. This capability makes CNN exceptionally well-suited for image classification and object detection tasks.

doi:10.20944/preprints202501.0521.v3

CNN consists of several essential components: convolutional layers, which extract local features and spatial hierarchies; pooling layers, which perform downsampling to reduce dimensionality; and fully connected layers, which aggregate global features and enable decision-making. To integrate these elements, CNN uses flattening to convert the outputs of convolutional and pooling layers into a one-dimensional vector, serving as input for the fully connected layers. The architecture prioritizes parameter sharing, enabling efficient processing of visual data.

Owing to its innovative design, CNN has become instrumental in advancing image processing. They are especially powerful in visual understanding because of their ability to extract and process spatial features. Its impact extends beyond image processing to object detection, image classification, and semantic segmentation tasks. In CPS which integrates computational algorithms with physical processes, CNN provides the robust perception capabilities necessary for effective environmental interaction, solidifying its role as a cornerstone of modern CV. Table 2 illustrates recent CNN techniques for CV applications.

Reference	CNN Techniques / Models	Key Contributions	Main Tasks / Application Domains	Major Limitations
[23]	U-Net architecture, Diffusion models, Optical flow estimation, Image-to-image models, and Frame interpolation	It manages imperfections in flow estimation effectively and decoupled edit propagate design	Local edits and short-video creation	Dependence on the first frame and struggle with highly complex or rapid motions
[24]	ResNet, Dense Networks (DenseNet), Generative Adversarial Networks (GANs) and Multi-scale Networks	Advancing techniques	Video object detection for medical imaging, surveillance, and autonomous driving	Artificial degradation may not apply to real-world situations.
[6]	Fully convolution neural network (FCNx) for classification tasks and ResNet for feature extraction	Hybrid multi-sensor fusion uses encoder-decoder FCNx with extended Kalman Filter for environmental perception.	Environmental perception for autonomous driving	Significant computational resources
[25]	CNNs various model compression techniques	Comprehensive review	Mobile devices, edge computing, IoT and embedded systems	A trade-off between computation and performance
[17]	CNNs on TensorFlow and TensorRT platforms via parallelism	Comprehensive Latency Analysis, Novel Measurement Techniques, optimisation Strategies, and Latency-Throughput Trade-Offs	Reducing latency in cloud gaming, optimising AR and VR delay applications and strategies to object detection and recognition models	Sensor Dependency and significant computational resources
[26]	Sparse Polynomial Regression and Energy-Precision Ratio (EPR)	Predictive Framework: NeuralPower	Mobile Devices, Data Centres, and embedded systems	Specific GPU platforms and may not generalize well to all hardware configurations
[27]	Mask R-CNN: Extends Faster R-CNN, Region of Interest (RoI) Align, FCNs for the mask prediction	Instance Segmentation, accuracy improvements in pose estimations	Object Detection and Segmentation, human pose estimations, AR applications	Significant computational resources, performance depending on specific applications and datasets
[28]	Single Shot MultiBox Detector (SSD), uses of default boxes and multiscale feature maps in detecting objects	Unified framework: SSD for real-time detection	real-time object detection for autonomous driving, embedded systems, and AR applications	Significant computational resources, not well performance on very small objects
[29]	DEtection TRansformer (DETR): Combines a common CNN backbone with a transformer architecture.	End-to-End Object Detection and Bipartite Matching Loss	Object Detection in various applications: autonomous driving, surveillance, and robotics.; and Panoptic Segmentation	Significant computational resources, not well performance on very small objects
[30]	Pre-trained CNN model for image extraction and Truncated Gradient Confidence-Weighted (TGCW) Model for online classification	Improved accuracy and efficiency by noise handling	Image classification in medical imaging and personal credit evaluation	Significant computational resources and noise sensitivity
[9]	Preprocessing step using OpenCV, YOLOv5 for real-time object detection	Integrated Framework for precise position estimation and error levels below 1 degree and 3D rendering of vehicles and their surroundings in digital twin visualisation	Accurate position estimations	inaccuracies in varying lighting or occlusion scenarios
[31]	3D Coordinate Mapping and Hybrid Reality Integration	Development of a Hybrid Reality-Based Driving Testing Environment	Autonomous Driving Development and extends to Internet of Vehicles	Reducing stability in higher frequency and incomplete real-world testing
[18]	Parallel light field platform, a data-driven approach for self-occlusion and inconsistency in viewpoints, colmap for offline re-construction	Improvements in PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure) metrics.	Applications requiring accurate 3D modelling and relighting, such as virtual reality, game development, and visual effects	Variations in colour temperature affecting 3D reconstruction and low-quality reconstruction models
[32]	Adaptive LFV coding and future integration with decentralized deep learning	balancing computation and communication latency to optimize performance	Enabling realistic digital twins, VR, AR and IoT-driven applications	Processing in off-line, not well performing in dynamic lighting conditions and occlusions
[33]	Integration of Yolov7 for human pose estimation and the DeepFace pre-trained model for age, gender, and race estimation	While Yolov7 performed well, the DeepFace model fell short in accuracy	The task of estimating human height from a single full-body image	Inaccurate performance in the DeepFace model and only single image input
[34]	EmoFusioNet, a deep fusion-based model	EmoFusioNet uses stacked and late fusion methods to ensure a color-neutral ER system, achieving high accuracy	A real-time facial emotion-based security	Underperformance for very dark-skinned individuals due to poor resolution of CMOS cameras

Table 2. A Table for CNN Techniques of Computer Vision Applications

Recent advancements in object detection and image classifications have focused heavily on different approaches, Region-Based Convolutional Neural Networks (R-CNN), Residual Networks (ResNet) and You Only Look Once (YOLO). These methods are often benchmarked against datasets like Microsoft COCO and ImageNet [35].

R-CNN is a two-stage object detection model. It generates around 2,000 region proposals per image, resizes each, and processes them through separate networks for feature extraction and classification [35]. To improve efficiency, regions with significant overlap are discarded, keeping only the highest-scoring classified regions. However, this approach is computationally intensive. To address this, Fast R-CNN and Faster R-CNN were developed to streamline the process, reducing processing time and improving accuracy.

Mask R-CNN, an extension of Faster R-CNN, adds a branch for instance segmentation, enabling the prediction of both bounding boxes and segmentation masks. This versatility allows it to handle tasks beyond object detection, such as human pose estimation while maintaining a relatively low computational overhead. Mask R-CNN operates at about 5 frames per second (fps) and is adaptable for other applications with minimal effort [27].

ResNet is a CNN architecture designed for feature extraction and image classification, with a primary focus on training deep neural networks efficiently without performance degradation, such as vanishing gradients. It employs residual learning with skip connections, enabling gradients to flow directly through the network. This innovation makes very deep networks, such as ResNet-50 and ResNet-101, both trainable and efficient. ResNet is widely used for tasks like image classification, image segmentation, and object detection, often serving as a backbone in detection models.

YOLO is a single-stage detector optimised for speed, making it ideal for real-time object detection. Unlike R-CNN, YOLO processes the entire image in a single pass through one network, generating fewer than 100 bounding box predictions per image [35]. Although faster, YOLO tends to have a higher localisation error than R-CNN but produces fewer background false positives.

Several enhanced versions of the YOLO architecture, including YOLOv2, YOLOv3, YOLOv4, and YOLOv5, have been introduced to improve accuracy while retaining the high speed required for real-time applications. Though generally less accurate than Faster R-CNN, these versions are fast enough to meet the demands of real-time systems such as self-driving cars [36].

Other models, such as the Single Shot Multibox Detector (SSD), have been proposed as alternatives to YOLO, offering improvements in the network's backbone structure [28]. Simultaneously, innovations like focal loss have been introduced to replace traditional loss functions, enhancing detection accuracy.

4.1.2. Federated Learning

Federated Learning (FL) holds significant promise for synchronising distributed CPS nodes because it trains models across multiple devices while keeping data localized. This approach enhances privacy and minimises the need for centralized data storage, a critical advantage for sensitive applications.

CPS typically operates through a network of distributed devices, such as sensors, actuators, and edge devices, spread across various physical locations. FL enables these devices to collaboratively train a shared model without centralising data. By aggregating model updates instead of raw data, FL supports decentralized architectures, aligning models across nodes while maintaining data privacy.

Given that CPS involves distributed components requiring seamless coordination, FL provides a privacy-preserving and decentralized mechanism to synchronize these components effectively. This ensures synchronized decision-making and consistent behaviour across the entire CPS network. FL also facilitates continuous learning, allowing devices to locally update models and periodically synchronize them. Such capabilities are crucial for real-time applications like autonomous vehicles and industrial robotics.

FL offers several advantages for CV applications in CPS [37]. The following are some key advantages.

- Privacy Preservation: FL retains data on local devices, sharing only model updates. This safeguards sensitive visual data, such as surveillance footage or medical records, addressing significant privacy concerns.
- Scalability: FL efficiently handles large-scale distributed systems, making it ideal for extensive CPS networks with numerous devices.
- Reduced Latency: Local data processing and updates minimise communication overhead and latency compared to centralized training methods.
- Heterogeneity Handling: FL can leverage adaptive aggregation techniques and personalized models to address the heterogeneity among nodes, ensuring synchronisation is maintained even in diverse and resource-imbalanced environments.

- Robustness and Adaptability: FL supports continuous learning and adapts to new data, enhancing the robustness of models in dynamic environments.

FL has two synchronisation techniques[38]. They are the following.

- Synchronous FL: All nodes synchronize their updates simultaneously, which can be challenging due to varying computational capabilities and network conditions.
- Asynchronous FL: Nodes update the model independently, offering more flexibility and efficiency but potentially leading to stale updates.

FL faces several challenges that researchers are actively working to address. Here are some of the key challenges.

- Non-IID Data: Data from different nodes may not be identically distributed, which can affect model performance. Techniques like data augmentation and domain adaptation can help mitigate this issue.[39]
- Communication Overhead: Efficient communication protocols and compression techniques are essential to reduce the bandwidth required for model updates.[37]
- Model Heterogeneity: Different devices may have varying computational capabilities. Federated learning frameworks need to account for this by using adaptive algorithms that can handle heterogeneous environments.[37]

FL plays a pivotal role in CPS synchronisation by facilitating decentralized collaboration, real-time adaptation, preservation of privacy, and scalability. It enables distributed devices to collaboratively train and synchronize models, effectively addressing CPS-specific challenges. This ensures efficient, reliable, and privacy-conscious coordination in modern smart systems.

4.1.3. Meta-Learning

Meta-learning in CV focuses on training models that can quickly adapt to new visual tasks with minimal data, computational effort, and dynamic scenarios. This is particularly useful in CV applications where tasks vary widely and data is scarce. Meta-learning techniques enable CV models to excel at tasks with very little labelled data, such as identifying new object classes from just a few examples. Meta-learned models can extract broadly applicable features, enabling rapid adaptation across diverse visual domains.

Meta-learning offers several techniques in the field of CV. Here are some key techniques.

- Prototypical Networks: These networks address the problem of few-shot classification by enabling generalisation to new classes with only a few examples per class. They learn a metric space where classification is based on distances to class prototype representations. They offer a simpler inductive bias compared to other few-shot learning methods, producing excellent results with limited data. [40]
- Siamese Networks: These networks consist of twin neural networks that share parameters and weights. They are trained to maximize the distance between dissimilar pairs and minimise the distance between similar pairs. which consists of twin networks with shared weights trained to map similar observations close together in feature space and dissimilar ones farther apart. Experiments on cross-domain datasets demonstrate the network's ability to handle forgery across various languages and handwriting styles. [41]
- Model-Agnostic Meta-Learning (MAML): The MAML algorithm is compatible with any model trained by gradient descent, applicable to tasks such as classification, regression, and reinforcement learning. The objective is to train a model on diverse tasks to generalize to new tasks with minimal training samples. This method optimises model parameters to enable rapid adaptation with just a few gradient steps on new tasks, making the model easy to fine-tune. MAML achieves state-of-the-art performance on few-shot image classification benchmarks, delivers strong results in few-shot regression, and accelerates fine-tuning in policy gradient reinforcement learning. [42]

- Memory-augmented models: These models, such as Neural Turing Machines (NTMs), can enhance the efficient incorporation of new information without relearning their parameters by quickly encoding and retrieving new information. They can quickly assimilate data and predict accurately with only a few samples. Santoro et al., 2016 [43] introduce a novel method for accessing external memory that focuses on memory content, eliminating the dependence on location-based mechanisms used in previous approaches.

Meta-learning offers several advantages in the field of CV. The following are some key benefits.

- Fast Adaptation: Meta-learning enables models to quickly adapt to new tasks with minimal data. It is critical for dynamic applications, such as autonomous vehicles or drones operating in changing environments.
- Data Efficiency: By leveraging prior knowledge from related tasks, meta-learning reduces the need for extensive training data. This efficiency is crucial in applications like medical imaging, where annotated data is often scarce.
- Cross-Domain Learning: Meta-learning helps models generalize better across different tasks and domains. That facilitates adaptation across domains, such as transferring knowledge from medical imaging to aerial imagery. Google Vizier includes features such as transfer learning, which allow models to use knowledge from previously optimised tasks to accelerate and enhance the optimisation of new ones[44].
- Personalisation: Meta-learning adapts models to individual preferences or environments, such as tailoring AR applications for unique users.

Meta-learning has numerous applications in CV to improve model performance and adaptability across various tasks. Here are some prominent examples.

- Image Classification: Meta-learning algorithms can quickly adapt to classify new categories of images with minimal data, and quickly recognize unseen classes in few-shot or zero-shot settings.
- Object Detection and Tracking: By leveraging prior knowledge, meta-learning models can enhance object detection and tracking capabilities, making them more robust to variations in the visual environment.
- Image Segmentation: Meta-learning can improve the performance of image segmentation tasks, where the goal is to partition an image into meaningful segments. This is particularly useful in medical imaging and autonomous driving.
- Facial Recognition: Meta-learning techniques can be used to develop facial recognition systems that adapt quickly to new faces with limited training data, enhancing security and personalisation applications.
- Pose Estimation: Meta-learning can be applied to pose estimation tasks, where the model needs to predict the pose of objects or humans in images. This is useful in the fields of robotics and augmented reality.
- Scene Understanding: Meta-learning allows CV systems to interpret new or unseen scenes for applications such as navigation or augmented reality (AR).

Meta-learning in CV faces several challenges that researchers are actively striving to overcome. Here are some notable challenges.

- Scalability: Meta-learning algorithms often struggle with scalability when applied to large-scale datasets and high-dimensional data typical for CV tasks. Efficiently scaling these algorithms while maintaining performance is a significant challenge.
- Generalisation: Ensuring that meta-learning models generalize well across a wide range of tasks and domains is difficult. Models trained on specific tasks may not perform well on unseen tasks, highlighting the need for better generalisation techniques.
- Computational Complexity: Meta-learning methods can be computationally intensive, requiring significant resources for training and adaptation. This complexity can limit their practical application, especially in resource-constrained environments.

- Data Efficiency: When meta-learning aims to be data-efficient, achieving this in practice can be challenging. Models often require a careful balance between leveraging prior knowledge and adapting to new data with minimal samples.
- Task Diversity: The diversity of tasks used during meta-training is crucial for the model's ability to generalize. However, creating a sufficiently diverse set of tasks that accurately represent real-world scenarios is challenging.
- Optimisation Stability: Ensuring stable and efficient optimisation during the meta-training phase is another challenge. Meta-learning models can be sensitive to hyperparameters and the choice of optimisation algorithms.
- Interpretability: Meta-learning models, especially those based on deep learning, can be difficult to interpret. Understanding how these models make decisions and adapt to new tasks is important for trust and transparency.

4.2. Themes and Categories

4.2.1. Synchronisation Strategies

Synchronisation refers to aligning the timing and interaction between various subsystems, sensors, and actuators within a CPS. In the context of ML-based computer vision, a list of synchronisation strategies is the following:

- Timestamping: Timestamping involves attaching precise time metadata to each data packet as it is generated, enabling the alignment and correlation of data streams from heterogeneous sources. Yang and Kupferschmidt [45] implement timestamp synchronisation specifically for video and audio signals, demonstrating its effectiveness. This approach is typically simpler and less computationally intensive compared to more complex synchronisation methods.
- Sensor Fusion: This technique is widely used in embedded systems to integrate data from multiple sensors, providing a more accurate and reliable representation of the environment. It is commonly applied in areas such as autonomous vehicles, robotics, and wearable devices. [**?**] introduce a real-time hybrid multi-sensor fusion framework that combines data from cameras, LiDAR, and radar to enhance environment perception tasks, including road segmentation, obstacle detection, and tracking. The framework employs a Fully Convolutional Neural Network (FCN) for road detection and an Extended Kalman Filter (EKF) for state estimation. Designed to be cost-effective, lightweight, modular, and robust, the approach achieves real-time efficiency while delivering superior performance in road segmentation, obstacle detection, and tracking. Evaluated on 3,000 scenes and real vehicles, it outperforms existing benchmark models.
  Moreover, Robyns et al.[9] demonstrate how to communicate from the physical system to the digital twin for visualising the industrial operation by using Unreal Engine. The digital twin features a modular architecture based on the publish-subscribe pattern, enabling the integration of multiple data processing modules from heterogeneous data streams.
- Real-time task scheduling: This technique involves orchestrating machine learning and computer vision tasks to ensure timely and reliable operations. CPS applications, such as autonomous vehicles, robotics, and smart manufacturing, demand low-latency, high-accuracy processing while operating under strict deadlines and resource constraints as illustrated in Figure 9.
  Inside the figure, it is often hard to neatly separate some machine learning techniques or technologies into just one category: Synchronisation, Optimisation, or Adaptation. Many methods operate across multiple dimensions simultaneously, especially as systems become more complex, distributed, and real-time.
- The Interplay Between Synchronisation, Optimisation, and Adaptation: This is pivotal for the seamless operation of ML-based CV in CPS.
  - Synchronisation and Optimisation: Efficient synchronisation reduces redundant computations and data transmissions, thereby optimising resource usage.

doi:10.20944/preprints202501.0521.v3

- Synchronisation and Adaptation: Timely synchronized data streams enable models to adapt quickly to environmental changes, enhancing responsiveness.
- Optimisation and Adaptation: Optimized models with reduced complexity facilitate faster adaptation to new data, ensuring real-time performance.

By integrating these components, CPS can achieve higher levels of autonomy, efficiency, and resilience, essential for applications like autonomous vehicles, smart manufacturing, and intelligent surveillance.

Techniques like federated learning in [46] have grown in scope, adopting strategies from all three areas. [46] proposed a reinforcement learning-based federated training scheme for object detection that involves synchronisation through the edge server's use of a shared Q-table, which coordinates and accelerates policy decisions across multiple mobile devices. This approach optimizes device selection and training data management to balance energy consumption, detection accuracy, and latency while reducing communication overhead in mobile edge computing. Additionally, the scheme adapts dynamically to varying channel conditions, prior training results, and the presence of jamming attacks by adjusting training policies on each device. This adaptability enhances the robustness and effectiveness of computer vision model training in challenging, real-world environments.

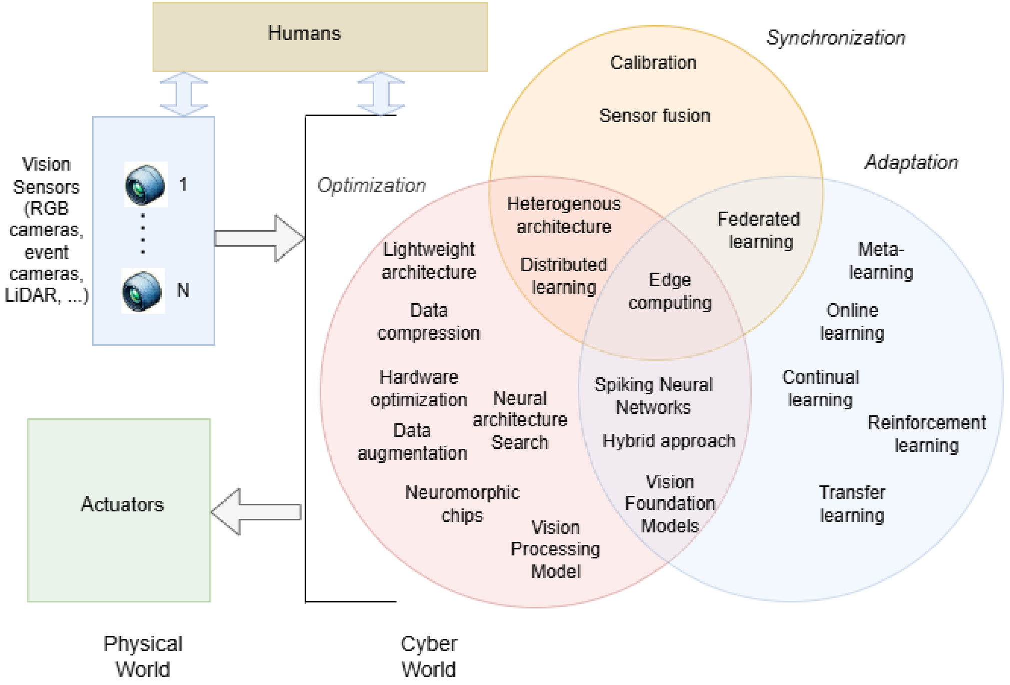

Figure 9. Framework for Synchronisation, Optimisation and Adaptation in Computer Vision in CPS

Hu et al.[47] propose a framework to enhance the efficiency of AI-based perception systems in applications like autonomous drones and vehicles. The framework focuses on prioritising the processing of critical image regions, such as foreground objects, while de-emphasising less significant background areas. This strategy optimizes the use of limited computational resources. The study leverages real LiDAR measurements for rapid image segmentation, enabling the identification of critical regions without requiring a perfect sensor. By resizing images, the framework balances accuracy and execution time, offering a flexible approach to handling less important input areas. This method avoids the extremes of full-resolution processing or completely discarding data. Experiments are conducted on an AI-embedded platform with real-world driving data to validate the framework's practicality and efficiency.

4.2.2. Optimisation Approaches

Balancing computational efficiency and accuracy is a critical challenge when applying ML techniques to CV within CPS. CPS systems are often constrained by limited computational resources (such as low-power embedded devices), real-time processing requirements, and the need for high accuracy in tasks like object detection, tracking, segmentation, and decision-making. Below are several optimisation approaches that can help strike a balance between these competing demands:

- Model Compression Techniques: Techniques [25] such as pruning, quantisation, knowledge distillation, low-rank factorisation, and transfer learning are applied to reduce the size of deep learning models without sacrificing significant performance. This is particularly critical for edge devices and CPS with limited hardware resources [13,48].
  - Pruning: Reducing the number of neurons or connections in a neural network by removing weights that have little influence on the output. This decreases the size of the model, making it computationally more efficient without significantly sacrificing accuracy.
  - Quantisation: Reducing the precision of the weights and activations in the model, from 32-bit floating-point to 8-bit integer or even binary. This leads to reduced memory footprint and faster computation, especially on specialised hardware (like FPGAs and GPUs).
  - Deep compression: Han, Mao, and Dally [15] introduce "deep compression", a three-stage pipeline (pruning, quantisation, and Huffman coding) designed to reduce the storage and computational demands of neural networks, enabling deployment on resource-constrained embedded systems. Pruning removes unnecessary connections, reducing the number of connections by 9× to 13×. Quantisation enforces weight sharing, reducing the representation of each connection from 32 bits to as few as 5 bits. Huffman coding further compresses the quantised weights. Experiments on AlexNet showed a 35× reduction in weight storage, with VGG-16 and LeNet achieving 49× and 39× reductions, respectively, while maintaining accuracy. This compression enables these networks to fit into the on-chip SRAM cache, significantly reducing energy consumption compared to off-chip DRAM access. The approach enhances the feasibility of deploying complex neural networks in mobile applications by addressing storage, energy efficiency, and download bandwidth constraints.
  - Knowledge Distillation: A process where a smaller, less complex "student" model learns to approximate the outputs of a larger, more complex "teacher" model. This can yield a more computationally efficient model with a similar accuracy. Hinton, Vinyals, and Dean [49] demonstrate the effectiveness of distillation, successfully transferring knowledge from ensembles or highly regularised large models into a smaller model. On MNIST, this method works well even when the distilled model's training set lacks examples of certain classes. For deep acoustic models, such as those used in Android voice search, nearly all performance gains from ensembles can be distilled into a single, similarly sized neural net, making deployment more practical. For very large neural networks, performance can be further improved by training specialist models that handle highly confusable class clusters. However, distilling the knowledge from these specialists back into a single large model remains an open challenge. This approach highlights the potential of distillation to balance performance and efficiency in machine learning systems.
  - Low-rank factorisation - This reduces the number of parameters in deep learning models by approximating weight matrices with lower-rank matrices. This technique helps compress models and speed up training and inference. Cai et al.[50] propose a joint function optimisation framework to integrate low-rank matrix factorisation and a linear compression function into a unified optimisation approach, designed to reduce the number of parameters in DNNs, computational and storage costs, while preserving or enhancing model accuracy.
  - Transfer learning is a machine learning method that involves reusing a model trained on one task to solve a related task. This approach allows the model to leverage its prior knowledge, enabling it to learn new tasks effectively even with limited data. In CPS applications, transfer

learning minimises the need for extensive manual labelling by transferring insights from similar domains. By utilising models pre-trained on large-scale datasets (e.g., ImageNet) as a foundation, transfer learning avoids the need for training from scratch. Fine-tuning only a few layers enables CPS systems to adapt quickly to new tasks or environments, significantly reducing computational costs.

- Lightweight Architectures: Use specialised architectures designed for efficiency while maintaining good accuracy. These include models like MobileNet and EfficientNet, which are designed to run efficiently on resource-constrained devices.

Bird et al.[16] explore unsupervised transfer learning between Electroencephalography (EEG) and Electromyography (EMG) using both MLP and CNN approaches. The models were trained with fixed hyperparameters and a limited set of network topologies determined through a multi-objective evolutionary search. Identical mathematical features were extracted to ensure compatibility between the networks. Their research demonstrates the application of cross-domain transfer learning in human-machine interaction systems, significantly reducing computational costs compared to training models from scratch.

- MobileNet is a class of efficient models designed for mobile and embedded vision applications. Howard et al.[51] utilise a streamlined architecture with depthwise separable convolutions to create lightweight deep neural networks. Two global hyperparameters are introduced to balance latency and accuracy, enabling model customisation based on application constraints. Extensive experiments show that MobileNets perform well compared to other popular models on ImageNet classification. Their effectiveness is demonstrated across diverse applications, including object detection, fine-grain classification, face attribute analysis, and large-scale geo-localisation.
- EfficientNets are a family of CNNs designed to achieve high accuracy with significantly improved computational efficiency. They were introduced as a solution to the challenge of scaling CNNs while balancing resource usage and performance. Tan and Li [52] propose a compound scaling method, a simple and effective approach for systematically scaling up a baseline CNN while maintaining efficiency under resource constraints. Using this method, the EfficientNet models achieve state-of-the-art accuracy with significantly fewer parameters and FLOPS, and high performance on both ImageNet and five transfer learning datasets, demonstrating their scalability and efficiency.

- Hardware Acceleration and optimisation: The method often involves leveraging parallelism (e.g., through graphics processing units (GPUs) or specialised hardware like tensor processing units (TPUs)) or optimising the inference pipeline to speed up processing as illustrated in Figure 10.

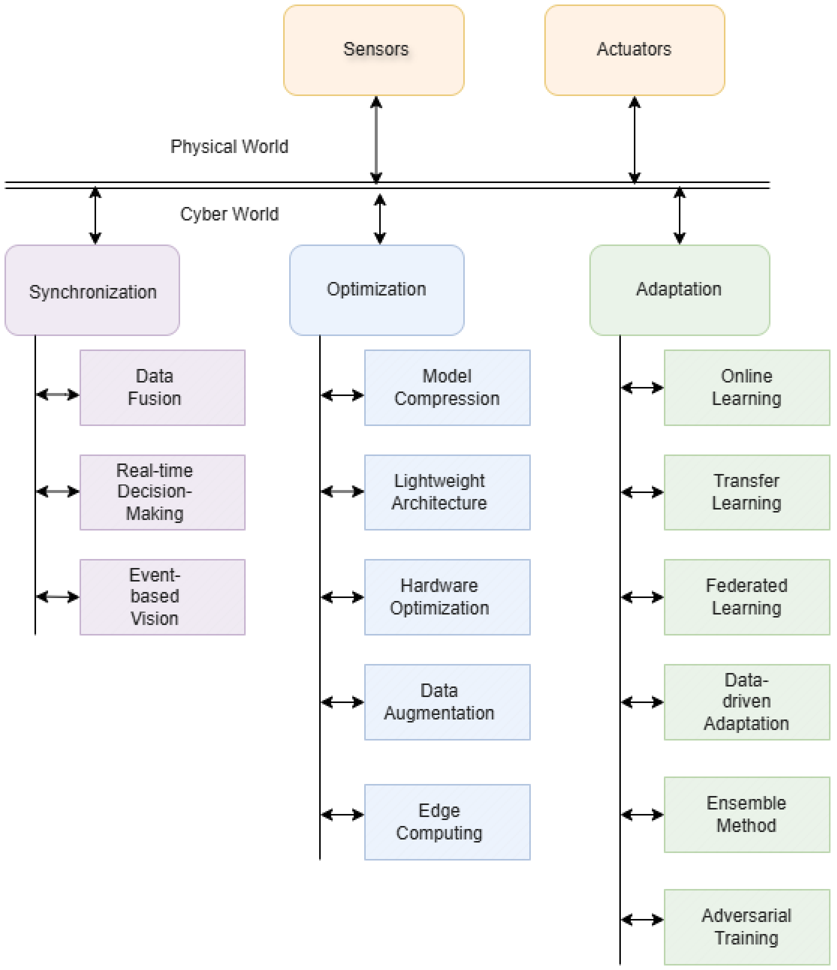

Figure 10. A Overview of the Proposed Framework.

Since 2015, distributed-memory architectures with GPU acceleration have become the standard for machine learning workloads due to their growing computational demands [48]. Maier et al. [11] depicts a GPU implementation of the parallel auction algorithm, optimised for both open computing language (OpenCL) and compute unified device architecture (CUDA) environments, which reduces memory usage and increases speed compared to previous implementations, making it ideal for embedded systems with large problem sizes. Experimental results across two GPUs and six datasets show a best-case speedup of 1.7x, with an average speedup of 1.24x across platforms. Additionally, this approach meets strict real-time requirements, especially for large-scale problems, as demonstrated in sensor-based sorting applications. However, optimisation is further constrained by fixed initial parameters, such as GPU architecture or model accuracy, limiting flexibility for future adjustments. Different GPUs deliver varied performance depending on factors such aslike batch size and execution context. Achieving optimal performance requires careful balancing trade-offs between accuracy, throughput, and latency [17].

To illustrate the trade-off between accuracy, latency, and throughput, we refer to the SNN performance with different time steps (T) with VGG16 on CIFAR-10 in [53]. Figure 11 shown in [53] shows the relationship between SNN accuracy and sparsity across various time window sizes for training the network for 100 epochs with a learning rate 1e-2. The figure demonstrates that increasing the time window size from 4 to 8 enhances the model's accuracy. A larger T allows neurons more time to integrate input spikes, leading to more precise outputs. This improvement in accuracy comes at the cost of increased latency, as the model requires more time steps to process each input. As latency increases with larger T values, throughput decreases. This inverse relationship means that while the model's accuracy increases, it processes fewer inputs in the same amount of time, which limits its usefulness in real-time applications. As a result, increasing

T improves accuracy, latency, and energy consumption; enhancing sparsity can mitigate energy costs but may affect accuracy. Therefore, optimising SNNs involves carefully tuning T and sparsity to meet specific application requirements for accuracy, throughput, and latency.

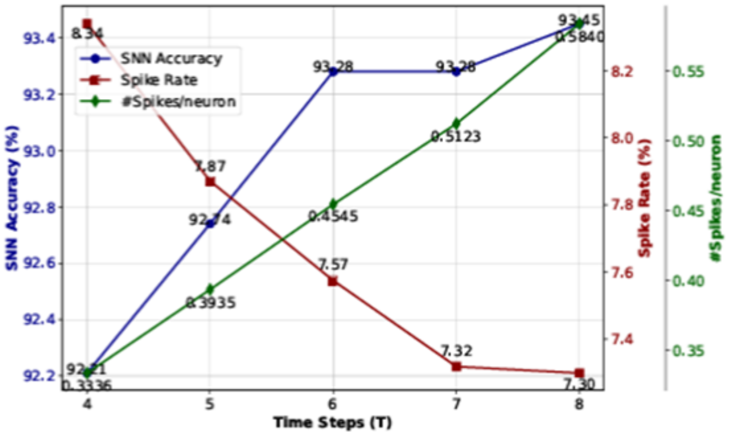

Figure 11. SNN performance with different T with VGG16 on CIFAR-10 [53]

- Data Augmentation: Data augmentation involves applying various transformations (such as rotation, scaling, and cropping) to the training dataset, thereby artificially expanding its size and diversity. This approach helps enhance the performance of smaller models. In many real-world scenarios, collecting sufficient training data can be challenging. Data augmentation [54] addresses this issue by increasing the volume, quality, and variety of the training data. Techniques for augmentation include deep learning-based strategies, feature-level modifications, and meta-learning approaches, as well as data synthesis methods using 3D graphics modelling, neural rendering, and generative adversarial networks (GANs).
  - Deeply Learned Augmentation Strategies: These techniques use deep learning models to generate augmentations automatically, improving the diversity and quality of the data. Neural networks are employed to create realistic data variations, thus enhancing the model's robustness.
  - Feature-level Augmentation: This method modifies specific features of the data, rather than the raw image itself. Common operations include changing attributes like contrast, brightness, or texture. Such adjustments can improve the model's ability to generalise across different scenarios.
  - Meta-learning-based Augmentation: Meta-learning approaches focus on learning how to generate useful augmentations based on the characteristics of the data. These methods aim to optimise the augmentation strategy itself, improving the model's learning efficiency across various tasks.
  - Data Synthesis Methods: These involve generating synthetic data through techniques like 3D graphics modelling. This approach creates realistic data variations, which is particularly useful for simulating rare or hard-to-capture events in real-world scenarios.
  - Neural Rendering: This technique uses neural networks to generate images from 3D models or abstract representations, producing realistic augmentations that can improve the diversity and realism of the training data.
  - Generative Adversarial Networks (GANs): GANs are employed to create synthetic data by training two competing networks—the generator and the discriminator. The generator produces new images, while the discriminator evaluates their authenticity. GANs can generate highly realistic augmentations, significantly boosting the dataset's diversity.
- Edge Computing: This paradigm involves moving computational tasks closer to the data source, such as on embedded devices at the network's edge. By processing data locally, edge computing

reduces the latency associated with transmitting data to and from remote servers, enabling real-time responses critical for applications like autonomous navigation and real-time surveillance. This approach also conserves bandwidth and enhances data privacy. Significant improvements in latency and throughput have been observed when deploying trained networks on mobile devices and remote servers [17].

Deng et al. [55] expand the scope of edge computing by integrating it with AI into a concept called Edge Intelligence, categorized into AI for Edge and AI on Edge:

- AI for Edge: Utilizes AI technologies to address key challenges in edge computing, such as optimising resource allocation, reducing latency, and managing data efficiently.
- AI on Edge: Focuses on performing the entire AI lifecycle, including model training and inference, directly on edge devices.

In distributed learning, the model is trained collaboratively across multiple edge devices, with only model updates—rather than raw data—being transmitted to a central server. This approach reduces communication bandwidth requirements and enhances data privacy. Tron and Vidal [56] demonstrate the application of distributed computer vision algorithms, highlighting that the storage requirements at each node depend solely on local data and remain constant irrespective of the number of cameras involved. For accelerating deep learning training, integrating distributed architectures with techniques such as gradient compression and adaptive learning rates is essential [14].

4.2.3. Adaptation Mechanisms

CPSs often operate in dynamic and unpredictable environments. Machine learning models must be adaptable to new conditions or evolving system requirements. Here are key adaptation mechanisms to ensure robust performance:

- Data-driven Adaptation: This approach involves leveraging data to enable models or systems to adjust and optimise their performance in response to dynamic conditions or specific challenges. In Shen et al.'s studies[18], the parallel light field platform supports the collection of realistic datasets that capture diverse lighting conditions, material properties, and geometric details. These datasets empower data-driven adaptation by providing models with inputs that closely mimic real-world scenarios, ensuring robust generalisation across varying environments. To handle self-occlusion, the conditional visibility module adopts a data-driven strategy, dynamically computing visibility along rays based on input viewpoints. Instead of relying on predefined rules, the module learns and predicts visibility directly from data, enabling it to adapt effectively to diverse viewing conditions. Moreover, data-driven techniques are applied to address specular reflection challenges and depth inconsistencies, showcasing the system's capability to adapt to complexities arising from changing viewpoints. These adaptations, powered by data, enhance the model's ability to adjust predictions under varying environmental and geometric conditions. Another example is presented in Kaur et al.'s article [19], where data augmentation techniques are used to generate variations in the dataset, allowing models to learn from a wide range of scenarios. This helps models adapt to unseen conditions during inference. The techniques discussed include Geometric Transformations, Photometric Transformations, Random Occlusion, and Deep Learning-based Approaches. The choice of augmentation methods depends on the nature of the dataset, the problem domain, and the number of training samples available for each class.
- Online Learning: This approach involves continuously updating a model with new labelled or pseudo-labelled data collected during deployment. In machine learning, models must learn and adapt in real time as fresh data becomes available. This is especially crucial in CPS where the system must adjust to changes such as varying lighting conditions for cameras or evolving cyber-security threats. Implementing online learning in production environments typically requires several steps: debugging offline, continuous model evaluation, managing data drift, performing

regular offline retraining, using efficient algorithms, ensuring data quality, having a rollback plan, and applying incremental updates [21].

For online learning, Hu et al. [30] introduce the pre-trained Truncated Gradient Confidence-weighted (Pt-TGCW) model, which combines offline and online learning techniques for tasks like image classification. This model highlights the effectiveness of incremental learning approaches. Additionally, Lu et al. [57] propose Passive-Aggressive Active (PAA) learning algorithms, which update models using misclassified instances and leverage correctly classified examples with low confidence. Their methods enhance performance across various online learning tasks, including binary and multi-class classification.

- Transfer Learning: This approach involves leveraging pre-trained models on large datasets and fine-tuning them for specific tasks, utilising existing knowledge to improve robustness. In CPS, models trained on one dataset may need to be adapted to different environments or contexts. TL enables this adaptation by fine-tuning pre-trained models with smaller, task-specific datasets, making it easier to adjust models to new situations. This is particularly important in CPS, where models must be trained in one context and then applied to another. For instance, Wang et al. [20] propose a transfer-learning approach for detecting attacks in CPS using a Residual Network (ResNet). Their method refines source model parameters through an intentional sampling technique, constructing distinct sample sets for each class and extracting relevant features from attack behaviours. This approach results in a robust network capable of accurately detecting attacks across different CPS environments.
- Ensemble Methods: The method combines multiple models to enhance prediction accuracy and reliability, addressing the weaknesses of individual models. The ensemble model proposed by Tahir et al. [22] incorporates diverse architectures (MobileNetV2, Vgg16, InceptionV3, and ResNet50), each capable of adapting to different features or patterns within the dataset. These models may excel in recognising distinct aspects of the data, and their combination allows the system to handle a wider range of scenarios and data variations, such as differences in X-ray image quality or fracture types. By aggregating predictions from multiple models, the ensemble approach adapts to changes in data quality and characteristics, improving robustness and generalisation. This is particularly important when working with medical datasets like Mura-v1.1, where data can vary in terms of noise, resolution, and imaging conditions. Preprocessing techniques such as histogram equalisation and feature extraction using Global Average Pooling further support adaptation, helping the model adjust to variations in image quality. These methods ensure that the model can effectively handle different input characteristics. The combination of diverse architectures and preprocessing techniques in the ensemble model enhances its adaptability, robustness, and accuracy, which is crucial for reliable performance in the complex and variable field of medical image analysis.
- Adversarial Training: This technique enhances the model's robustness by making it more resistant to small, intentional perturbations in the input data that could otherwise lead to misclassifications. By generating adversarial examples [58] and incorporating them into the training process, the model learns to recognize and correctly classify inputs that would typically confuse it, thus improving its generalisation capability. This approach provides insights into how neural networks can adapt to better resist adversarial perturbations, ultimately strengthening their robustness. By using adversarial examples during training, the model becomes more adaptable to a wider range of input variations, making it more resilient and capable of generalising effectively across different datasets, architectures, and training conditions.

  Another example [59] involves handling adversarial perturbations through randomized smoothing, which strengthens a model's robustness against adversarial attacks by adding Gaussian noise to the input data. This technique ensures the model is "certifiably robust" to adversarial perturbations, enabling it to maintain reliable performance even when confronted with modified inputs. Training the model with both original and noise-augmented data enhances its capacity to generalize across varied conditions, including adversarial scenarios. This adaptation process equips the model to handle a broader range of input variations, increasing its resilience to unforeseen changes in data distribution. As a formal adaptation technique, randomized smoothing ensures stability and high performance, even under adversarial conditions. By incorporating noise during training, this method significantly bolsters the model's ability to manage adversarial inputs, enhancing its robustness and generalisation in challenging environments.

- Federated Learning: In distributed CPS, where devices are spread across different locations (e.g., smart cities, industrial IoT), FL allows individual devices to train models locally and share updates, improving model performance across the system without centralising sensitive data.
  In Himeur's article [37], FL is used to distribute computational tasks across multiple clients, alleviating the load on central servers and enabling collaborative machine learning while ensuring data privacy. FL employs various aggregation methods, such as averaging, Progressive Fourier, and FedGKT while incorporating privacy-preserving technologies like Secure Multi-Party Computation (MPC), differential privacy, and homomorphic encryption to safeguard sensitive information. Despite its advantages, FL in Computer Vision (CV) encounters several challenges, including high communication overhead, diverse device capabilities, and issues related to non-IID (non-independent and identically distributed) data, complicating model training and performance consistency.
  To lower resource constraints, Jiang et al. [60] introduce a Federated Local Differential Privacy scheme, named Fed-MPS (Federated Model Parameter Selection). Fed-MPS employs a parameter selection algorithm based on update direction consistency to address the limited resource issue in CPS environments. This method selectively extracts parameters that improve model accuracy during training while simultaneously reducing communication overhead.

4.3. A Practical Case Study - A Smart Surveillance System

4.3.1. Motivation

This subsection reviews a proposed framework that captures the evolution and current capabilities of a modern surveillance system designed to monitor environments in real time for security, safety, and operational intelligence. These systems are increasingly deployed across a variety of domains, including public spaces, transportation hubs, industrial facilities, and smart cities. In the context of this study, the focus is on an open car park with approximately 50 parking spaces within a university. For such a system to function effectively and ethically, it must address a range of technical, operational, and societal challenges. [61]:

- Real-Time Distributed Architectures: Modern surveillance often requires rapid decision making in multiple locations. Low-latency data collection, real-time processing, and communication in a distributed system require robust coordination between edge devices, cloud infrastructure, and control units, especially under bandwidth and resource constraints.
- Awareness and Intelligence: Surveillance systems must move beyond passive monitoring to active interpretation of scenes. This includes contextual understanding, anomaly detection, and behaviour prediction using AI and ML. The challenge lies in integrating these intelligent capabilities without overwhelming computational resources or compromising privacy.
- Video Analysis Limitations: Traditional video analytics struggles in low-light conditions, high-speed scenarios, or scenes with occlusions and noise. They also often rely heavily on high-bandwidth video streams. Overcoming these issues requires more adaptive sensing techniques, such as event-based cameras or multimodal sensor fusion.
- Energy Efficiency in Remote Sensors: Designing energy-efficient sensing, computation, and communication protocols is essential for sustained deployment without frequent maintenance.
- Scalability: As surveillance networks grow in size and complexity, maintaining consistent performance, data integrity, and manageability becomes increasingly difficult. Scalable architectures

must support the seamless integration of new sensors, load balancing, and adaptive processing without introducing bottlenecks.

Referring to Figure 8, the optical detection system comprises event cameras, RGB cameras, and LiDAR, which form a complementary multimodal sensing strategy. Each sensor modality contributes distinct, non-overlapping information. For example, RGB cameras capture rich color details, LiDAR provides precise spatial depth, and event cameras offer a high temporal resolution. When these data streams are effectively fused, they significantly enhance the robustness and accuracy of perception systems, particularly in challenging conditions such as complete darkness or heavy rain.

4.3.2. Physical Entities

- RGB Cameras are the core of a visual monitoring system. Its clear and easily understood images are helpful, especially those with high resolution and fast frame rates. However, its performance drops significantly in challenging situations like nighttime, rain, fog, or snow.
- Event Cameras are bio-inspired sensors that detect brightness changes asynchronously at each pixel. Each pixel operates independently, continuously monitoring the scene and triggering an event whenever the brightness change exceeds a predefined threshold. This results in a continuous stream of events, providing a dynamic, sparse representation of visual information [62]. Compared to conventional cameras, event-based sensors offer several advantages, including high temporal resolution, wide dynamic range, low latency, low power consumption, and reduced motion blur—beneficial for tasks such as object reconstruction, segmentation, and recognition [62–65]. However, event cameras can be sensitive to noise, especially in low-light situations. In fast-changing scenes, they can produce a very high number of events, leading to large data loads. Snowflakes or raindrops can create many false events, making it hard to detect real moving objects [66].
- LiDAR (Light Detection and Ranging) is a sensing technology that uses laser pulses to measure distance and create detailed 3D maps of the environment. When combined with cameras in surveillance systems, LiDAR adds precise depth information to the colour and detail that cameras provide. This combination improves how we see, understand, and respond to what's happening in an area.

  The main benefits of using LiDAR with cameras are as follows [**?**]:

  - Better 3D awareness: LiDAR accurately measures depth, helping build a clear 3D map. When combined with camera images, this gives a more complete picture, making it easier to detect and understand what's in the scene.
  - Improved Object Recognition: By using both LiDAR's shape and distance data along with a camera's visual detail, it becomes easier to tell different objects apart and reduce false alarms.
  - Works in the Dark and Bad Weather: LiDAR doesn't rely on light, so it works well at night or in dark areas. It also performs better than cameras in fog, smoke, or rain.
  - Smart Alerts: LiDAR can be used to set up virtual fences or tripwires. If something crosses them, the system can send an alert automatically.

  However, in low-visibility conditions, such as fog, LiDAR measurements degrade more than those from regular cameras because the laser pulses travel twice the distance, increasing the chances of scattering and attenuation.
- Computer Systems: A system can leverage parallelism through graphics GPU/TPU/optimising for both OpenCL and CUDA environments to reduce memory usage and increase speed, making it ideal for embedded systems. [11,48] The operating system used is Ubuntu 24.04 LTS with the popular deep learning framework PyTorch.
- Outputs: Visualisation screens

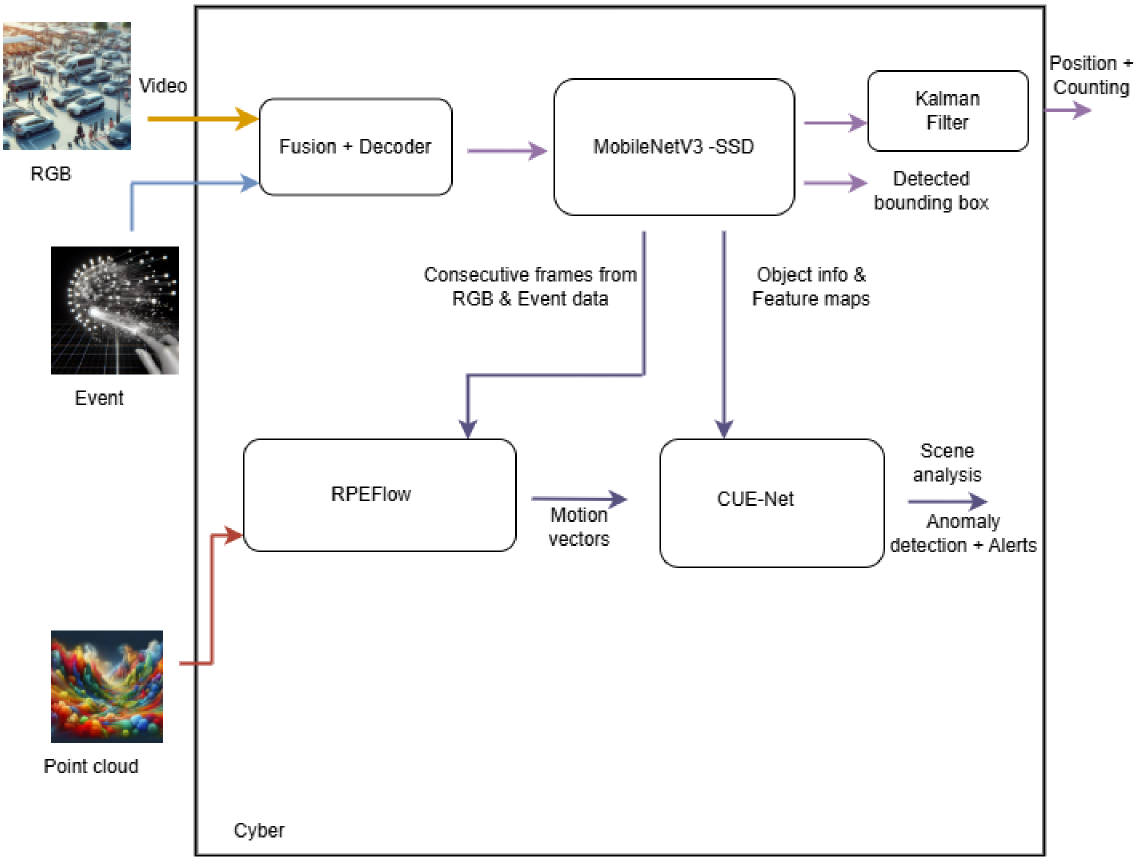

Figure 12. Proposed Architecture Design

4.3.3. Cyber World

- Data Fusion and Synchronisation
  - RGB camera-LiDAR calibration: The calibration is achieved by identifying the extrinsic parameters that maximize the mutual information between the two modalities. This optimisation can be carried out using standard tools in SciPy [66].
  - Event camera-LiDAR calibration: [67] proposed a novel method to calibrate the extrinsic parameters between a dyad of an event camera and a LiDAR without the need for a calibration board or other equipment. From the event camera, edges are obtained by accumulating events over time and applying edge detection filters. From LiDAR, geometric edge features are extracted by identifying sharp changes in depth (e.g., using surface normals or curvature). The method matches the edge features across the two modalities (event images and LiDAR projections) based on spatial alignment. The objective is to find the extrinsic parameters that best align the edges of both sensors.
  - Extrinsic calibration: While the RGB-LiDAR and event-LiDAR calibrations can be performed separately, an additional constraint is introduced to enforce consistency by incorporating the extrinsic transformation between the event camera and the RGB camera. This transformation is obtained through standard stereo calibration techniques, such as those available in OpenCV. By using a checkerboard visible in both cameras' fields of view, we estimate their relative pose by minimising the reprojection error. The resulting extrinsic are then applied as a constraint in a joint mutual information (MI) optimisation framework, where we aim to maximize the MI between the aligned sensor data [66].
  - RGB Cameras: [68] proposed a novel subframe synchronisation technique for multicamera systems by interpolating between frames, allowing the system to estimate what a camera would have captured at any precise point in time, even between actual recorded frames. This

ensures much tighter temporal alignment across cameras, which is critical for applications like 3D reconstruction, motion capture, and scientific analysis of fast phenomena.

  - Synchronisation: LiDAR is synced using PTP with software timestamping. Similarly, the RGB camera is programmed to be a PTP slave and additionally sends a trigger signal to the event camera at the start of data acquisition. The event camera receives the synthetic event from the RGB camera, which marks at the start time [66].

- Architecture: The proposed architecture is shown in Figure 12.
  - MobileNetV3: It serves as a robust and efficient backbone architecture for the surveillance system. Its lightweight design, incorporating depthwise separable convolutions and NAS, makes it highly suitable for real-time inference on edge devices [69]. When integrated with SSD, MobileNetV3 enhances object detection by combining its high-level feature extraction with SSD's fast, one-shot bounding box prediction, ensuring speed and accuracy [70,71].
  - Real-time Pixel-wise Estimation Flow (RPEFlow): To support reliable object tracking across video frames, RPEFlow is integrated into the detection pipeline because it provides dense motion vectors that help maintain object identity through occlusions or rapid movements, enabling consistent object association and precise motion trajectory estimation [72].
  - Contextual Understanding and Enhancement Network (CUE-Net): It is employed for anomaly detection, leveraging spatial appearance and temporal motion cues. It receives fused inputs from MobileNetV3-SSD detections and RPEFlow's motion stream to detect irregular behaviours such as sudden movements, crowding, or violent actions. Its ability to process spatiotemporal features enables real-time recognition of abnormal activities, enhancing situational awareness and proactive threat response [70].
  - SSD: The bounding boxes produced by SSD undergo post-processing via a Kalman filter bank. This lightweight and efficient tracker predicts future object positions based on motion dynamics, manages occlusions and missed detections through probabilistic estimation, and reduces false positives, resulting in smoother and more reliable object tracking [71].
- Optimisation
  - MobileNetV3: It incorporates several key optimisations to enhance performance in resource-constrained environments. It utilizes depthwise separable convolutions to significantly reduce computational load, squeeze-and-excitation (SE) modules to recalibrate channel-wise feature importance, and NAS to balance accuracy and latency effectively for real-time applications [69]. RPEFlow contributes by generating dense optical flow through real-time pixel-wise motion estimation, optimized for the temporal domain. It is designed for low-latency execution on edge devices, requiring minimal computational resources due to its reduced parameter footprint [72].
  - SSD: It enhances object detection by dividing the input image into a grid and applying multiscale default anchor boxes to capture objects of various sizes. Through its convolutional layers, SSD predicts both the object class and its bounding box location in a single forward pass, ensuring fast detection [71].
  - Kalman filter: It optimizes the tracking algorithm by predicting future positions of detected objects based on their velocity and motion patterns. It also helps reduce false positives and maintain accurate object tracking, especially during occlusion or brief detection failures [72]. Additionally, a scalable edge-to-cloud architecture allows models to run efficiently on edge devices for real-time operation while leveraging cloud resources for periodic retraining and long-term performance optimisation.
- Adaptation
  - MobileNetV3 + SSD: MobileNetV3 can be pre-trained on large-scale general-purpose datasets and subsequently fine-tuned using domain-specific surveillance footage to improve object recognition accuracy within targeted environments. The model supports dynamic input

scaling and incremental retraining, enabling it to adapt over time to new object classes, evolving visual patterns, or changing surveillance contexts. When combined with the SSD, MobileNetV3 supports adaptive object detection by leveraging SSD's default anchor boxes at multiple scales and aspect ratios, effectively identifying objects of various sizes in complex scenes [71]. SSD's performance can be further refined through feedback loops or active learning mechanisms, allowing the system to improve detection accuracy with minimal manual supervision.

- RPEFlow: It plays a crucial role in motion analysis by providing real-time, frame-wise dense optical flow, which is particularly useful in scenarios involving variable lighting, occlusions, or low visibility [72]. Its adaptive capability ensures robust motion estimation and consistent object tracking even under dynamic environmental changes. In parallel, CUE-Net is designed for anomaly detection, learning appearance features and motion dynamics to identify irregular or suspicious behaviours in surveillance footage. It employs weakly supervised learning, minimising the need for extensive labelled datasets, and supports domain adaptation to maintain performance across different surveillance settings.
  At the system level, machine learning adaptation is further enhanced through sensor fusion strategies. Inputs from RGB cameras, LiDAR, and event-based sensors are dynamically weighted based on environmental conditions [72]. For example, LiDAR and event data during nighttime or in low-light situations are highly preferable. Temporal adaptation is also implemented, with the system learning behavioural baselines over time and adjusting anomaly detection thresholds accordingly.

4.3.4. Evaluation Metrics

Despite the growing interest in applying ML to CV within CPS, there is currently no standardized procedure universally accepted for evaluating such models. This absence of standardization presents significant challenges in objectively assessing and comparing ML-based surveillance systems within CPS contexts. In response to the design criteria outlined in our motivation, we propose a multi-faceted evaluation framework tailored to ML-based surveillance systems, with a particular focus on their integration into CPS environments. This approach is informed by reviewing some relevant articles [73–77], and is intended to provide a structured basis for a comprehensive and consistent assessment.

- Object and anomaly detections: To evaluate these detection capabilities of the CV, we utilize a set of standardized quantitative metrics in conjunction with benchmark datasets such as MS COCO and OpenImages. The core evaluation metrics are as follows: [78,79]
  - Precision: It measures how many of the predicted positive detections (e.g. detected objects) are correct. It is defined as:

    $$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{TP + False Positives (FP)}}$$

    High precision indicates a low false positive rate. In surveillance, this means fewer false alarms from incorrectly detected objects.
  - Recall: It quantifies how many of the actual positives are correctly detected by the model. It is defined as:

    $$\text{Recall} = \frac{\text{TP}}{\text{TP + False Negatives (FP)}}$$

    High recall ensures the system misses as few real objects as possible.
  - F1-Score: It is the harmonic mean of precision and recall, balancing both metrics:

    $$\text{F1} = 2\,\text{x}\frac{\text{Precision x Recall}}{\text{Precision + Recall}}$$

It is especially useful when there is an uneven class distribution or when both false positives and false negatives are important.

- Accuracy: It shows how well the model performs. It is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- Intersection over Union (IOU): It is used to measure the overlap between a predicted bounding box and the ground truth box:

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

A higher IoU value means the prediction is more accurate. In evaluation, a threshold (>0.5) is often defined as a "correct" detection.

- Mean Average Precision (mAP): It summarises the precision-recall curve across different IoU thresholds and object classes. It is computed as:

$$\text{mAP} = \frac{1}{\text{N}} \sum_{i=1}^{N} AP_i$$

It reflects both the localisation and classification performance of an object detection system.

- System-level performance
  - Latency Measurement: Latency is defined as the time elapsed between the acquisition of an input (e.g., an image, video frame, or event stream) and the generation of the corresponding system output (such as object detection, anomaly alert, or visual feedback). It is a critical metric for evaluating the responsiveness of real-time surveillance systems. Latency can be measured in two main ways: End-to-end latency, which captures the total time from sensor input to final system output; and component-level latency, which isolates the processing time of individual modules (e.g., sensor capture, preprocessing, inference, and output rendering). Accurate measurement of both types helps identify bottlenecks and optimize system performance [69].
  - Processing Speed: Processing speed refers to the rate at which the surveillance system can handle input data, typically measured in frames per second (FPS) for image and video streams or points per second for point cloud data. This metric indicates the system's capacity to maintain real-time operation under varying workloads. [80] demonstrated a comparative implementation of processing speed in different perception models, highlighting its importance in evaluating real-time performance for safety-critical applications such as autonomous driving and surveillance.
  - Throughput: It refers to the total volume of data that the surveillance system can process per second. Reflects the overall data handling capacity of the system under operational conditions. [72] demonstrated throughput-aware optimisation for multisensor systems.
- Scalability: It refers to the system's capacity to sustain acceptable performance as deployment parameters expand, such as the number of integrated sensors, the size of the monitored area, the volume of concurrently tracked objects, or increasing computational demands.
  - Sensor Scalability: To assess sensor scalability, we progressively increase the number of input sources (ranging from 2 to 20 cameras or LiDAR units) and measure key performance indicators such as system latency, frame rate (frames per second), and throughput under each configuration. This evaluation reveals the system's capability to handle additional sensory input without significant degradation in performance.

  - Network Scalability: It is evaluated by monitoring the system bandwidth usage as the data transmission scales of the sensor. Tools such as NetHogs can be used to analyze real-time bandwidth consumption as demonstrated in [81].
- Robustness: It assesses the system's ability to sustain functional performance under adverse or variable conditions, such as poor lighting, sensor degradation, network interruptions, or adversarial inputs.
  - Synchronisation Robustness: It is primarily designed to test the synchronisation between two video sequences. The method aims to calculate a matching score between 0 and 1 for each pair of frames from two videos, creating a matching matrix (Matching Frame) and (Delay Estimation) using the matching matrix to estimate the delay between the video sequences by analyzing the pattern of highest matching scores across frames in [82].
  - Corruption Robustness: To enhance resilience against visual corruptions, two augmentation strategies are applied: (1) replacing each training image with a stylized version; and (2) augmenting the dataset by including stylized variants alongside the original images. This approach, shown to improve robustness in object detection tasks, follows the methodology outlined in [75,83].
  - Robust Perception Under Adverse Conditions: To evaluate perception robustness in challenging environments (e.g. rain, fog, and low lighting), the following procedures in [84] are adopted: (1) Data Augmentation via Unpaired Image-to-Image (I2I) Synthesis; (2) Two-Branch Architecture Design: Implement a generalized two-branch network that processes both the original and the I2I-enhanced images; (3) Comparative Analysis: Conduct a comprehensive performance analysis across three configurations—image enhancement only, augmentation only, and the combined two-branch architecture—to assess their effectiveness in improving robust perception.
- Security Evaluation: It involves a comprehensive assessment of how effectively the system safeguards against unauthorized access, data breaches, and malicious manipulation.
  - Physical Security Measures: It is the first line of defence and includes robust housing for sensors, cables, and computing infrastructure to protect them from environmental factors and tampering. Access control must be enforced through controlled physical access zones, complemented by visible signage to deter unauthorized entry and ensure clear surveillance coverage.
  - Data Protection and Privacy: The system must implement role-based access control to restrict sensitive data access to authorized personnel only. Multifactor authentication adds an essential layer of security for system access. Sensitive data should be securely stored, managed according to defined retention policies, and deleted when no longer needed. To protect the integrity and confidentiality of data in transit, end-to-end encryption should be adopted across video streams, sensor data, and command signals. Audit logs should be maintained and regularly reviewed to detect and respond to unauthorized access attempts.

5. Discussions

5.1. Critical Evaluation

5.1.1. Increased Focus on Real-Time Performance

- Synchronisation of Data Streams
  - Time Synchronisation: Accurate time synchronisation with stringent latency is essential for real-time CV applications in CPS, where multiple sensors, devices, and processing units must operate cohesively. The Flooding Time Synchronisation Protocol (FTSP) [85] is designed for distributed environments in real-time, aiming to minimise synchronisation errors between multiple nodes and achieve tighter timing accuracy. However, this approach

can introduce significant communication overhead, leading to network congestion and increased energy consumption.
To address these challenges, [12] demonstrates that an event-based consensus clock synchronisation scheme incorporating a low pass filter and a novel event-triggered mechanism can effectively reduce communication overhead. Furthermore, [86] explores various asynchronous time synchronisation protocols across different applications to improve FTSP scalability, adaptability, latency, and accuracy.
Moreover, integrating FTSP with techniques such as data compression, prediction algorithms, and data aggregation, can further improve its efficiency [87].
For wired connection networks, IEEE 1588 Precision Time Protocol (PTP) is specifically designed for high-precision synchronisation in industrial automation and robotics. [88] demonstrates that a multi-domain PTP system design can effectively mitigate network faults, ensuring reliable real-time control.
For lightweight synchronisation algorithms in CPS applications, the Collaborative Siamese Network (CoSiNeT) is a state-of-the-art method for handling network packet delay variations in industrial environments with constrained hardware and software resources [89].

  - Real-time Data Fusion: [6] demonstrates that a fusion framework combining FCNx with EKF provides a cost-effective, lightweight, modular, and robust solution for real-time road detection, including road segmentation, obstacle detection, and tracking in autonomous vehicles equipped with LiDAR, camera, and radar sensors. In industrial applications such as recycling, mining, and food processing, the auction algorithm [11] outperforms traditional multi-target tracking methods for sensor-based sorting.
  - Edge Computing: Recent advancements in edge computing focus on processing data locally to reduce latency and enhance real-time inference. However, operations across distributed nodes can lead to inconsistencies due to time drift. To address this, time synchronisation-aware edge-end collaborative network routing management (TSA-RM) in [90] optimizes routing to minimise the weighted sum of the model training loss function and delay. This approach ensures coherent and timely data processing while balancing training loss and latency.
    Balancing real-time performance with energy efficiency remains a key challenge for embedded vision hardware. Research in [91] on energy-aware edge computing indicates that offloading computational tasks to edge servers can significantly reduce power consumption while maintaining low latency. This strategy is particularly advantageous for mobile and embedded devices in autonomous systems, where energy resources are constrained.

- Optimisation of ML Models
  - Model Compression: Real-time optimisation techniques are designed to reduce computational demands while addressing the scalability and complexity of neural networks. Model compression methods [15,49,50] have been effective in simplifying neural architectures, while lightweight architectures [51,52] provide efficient solutions for real-time inference.
  - Hardware acceleration: Specialized hardware accelerators, including parallel and distributed capabilities, are pivotal for enhancing performance and efficiency. GPU-based implementations [11,17] optimize performance by minimising latency and increasing throughput. As scalable ML models evolve, the shift toward specialized hardware has accelerated. TPUs, designed for tensor operations, optimize large-scale, low-precision computations, while FPGAs offer reconfigurable logic for real-time tasks in industrial automation and robotics. ASICs provide ultra-low latency and high throughput for dedicated applications but lack flexibility and require high initial investment.
    Each hardware accelerator presents trade-offs in throughput, precision, power consumption, and area [92]. The industry is transitioning beyond GPU-centric approaches, leveraging a diverse array of specialized chips. Approaches like COSMOS (Coordination of High-Level Synthesis and Memory Optimisation for Hardware Accelerators) utilize Pareto-optimal im-

plementations to balance these trade-offs, reducing computational overhead and optimising energy efficiency [93].

  - Other Techniques: Edge computing reduces latency by processing data locally, minimising the dependence on centralized servers while enhancing efficiency and data privacy [55,94]. Real-time data augmentation dynamically transforms data during preprocessing, optimising it for run-time processing[19,54].

    To further reduce computational overhead in video processing, frame-skipping techniques selectively process key frames instead of running deep learning models on every frame. This leverages the temporal correlation between consecutive frames, ensuring minimal performance loss while significantly improving efficiency. For instance, FrameHopper [95] employs a reinforcement learning-based approach to determine optimal frame skip lengths, balancing detection accuracy with reduced processing load.

    Similarly, adaptive sampling strategies dynamically adjust the sampling rate based on the complexity of the scene or the motion patterns. Methods like MGSampler [96] optimize sampling by prioritising frames containing significant changes, reducing redundant computations while preserving critical information for object detection and tracking. These techniques collectively improve real-time performance in resource-constrained environments, making them vital for edge-based computer vision applications in CPS.

- Adaptation to Dynamic Environments
  - Model-Agnostic Meta-Learning (MAML): It is a meta-learning technique designed to enable machine learning models to adapt quickly to new tasks with minimal training data, particularly effective in few-shot image classification and object detection [42].
  - Domain Adaptation: In many CPS applications, the visual environment is non-stationary. For instance, an autonomous vehicle must handle variations in lighting and weather. A domain-adaptive object detection framework in [97] can address performance degradation in adverse weather under foggy and rainy conditions. The approach includes image-level and object-level adaptations to minimise domain gaps in image style and object appearance. Additionally, a novel adversarial gradient reversal layer enhances model robustness by mining hard examples, while an auxiliary domain through data augmentation enforces domain-level metric regularisation.

    [98] introduces the Gated Image-Adaptive Network (GIANet), a novel framework that enhances low-light images and clusters bounding boxes effectively.
  - Online Continual Learning: This approach allows models to adapt to new tasks or data distributions without catastrophic forgetting. Methods such as Elastic Weight Consolidation (EWC) [99,100] or memory replay mechanisms [101] help preserve previously learned knowledge while integrating new information, developing models capable of real-time adaptation.
  - Transfer Learning: Fine-tuning pre-trained models on a small subset of new data helps the model quickly adapt to new tasks or environments. Techniques such as few-shot learning can be particularly valuable when only limited new data is available.

5.1.2. Hybrid Approaches

- Digital Twin (DT): This approach integrates physical and virtual optimisation layers that emerge as powerful approaches to improve system efficiency and robustness in CPS. These methods leverage the strengths of both the physical domain (e.g., real-world sensors, actuators, and processes) and the virtual domain (e.g., simulations, predictive algorithms, and digital twins) to create a cohesive and adaptive system.

  A bio-inspired LIDA (Learning Intelligent Distribution Agent) cognitive-based DT architecture [102] facilitates unmanned maintenance of machine tools by enabling self-construction, self-evaluation, and self-optimisation. This architecture provides valuable insights into implementing real-time monitoring in dynamic production environments.

In the manufacturing industry, DT enhances flexibility and efficiency while addressing safety and reliability challenges in collaborative tasks between human operators and heavy machinery. They enable accurate detection and action classification under diverse conditions, as demonstrated in studies [9,12]. Another prominent application involves an Autonomous Driving test system under hybrid reality [31], which improves efficiency, reduces costs, and enhances safety, offering a robust solution for autonomous driving development.

However, the continuous data transmission from the physical world to the virtual DT system must demand extensive wireless resources. A continual reinforcement learning algorithm in [103] can learn a stable policy across historical experiences to quickly adapt to physical states and network capacity dynamics, optimising the digital twin synchronisation.

Nevertheless, the current state of DT technology often requires offline system halts for model updates, with implementations depending on backends that enforce strict data exchange constraints. To overcome these limitations, the CoTwin framework [104] introduces a dynamic approach that enables online model refinement in CPS without operational disruptions. By leveraging a blockchain-based collaborative space for secure data management and integrating neural network algorithms for fast, time-sensitive execution, this framework ensures stable and efficient performance. Additionally, it meets the stringent temporal requirements of CPS, providing a significant advantage in industrial applications.

- Hybrid Optimisation Algorithms: Integrating classical optimisation techniques with cutting-edge ML models offers a powerful approach to tackling complex challenges in CPS. For example, real-time tracking demands significant computational resources, posing challenges for deployment in resource-constrained environments. [105] addresses this by combining a classical optical flow algorithm with a deep learning model, striking a balance between accuracy and efficiency. Designed for human-crowd tracking, this method effectively reduces computational costs while maintaining high tracking performance.

  Similarly, [106] presents a hybrid approach for real-time object tracking and classification in real time, merging frame-based and event-based methods. This strategy enables real-time performance on embedded platforms without compromising accuracy, making it well-suited for energy-efficient applications.
- Integration with Transformer: Transformers utilize self-attention mechanisms [107] to capture global contextual information in images, unlike CNNs, which process them locally. With compression techniques and hardware acceleration, transformers can achieve faster and more efficient decision-making in resource-constrained environments. For instance, [108] optimizes the ARM Keyword Transformer (KWT) on a RISC-V platform, achieving a 5× speedup and power reduction for edge devices. Similarly, [109] employs distinct quantisation and approximation strategies in Softmax and LayerNorm, enhancing energy and area efficiency for end-to-end inference on GPU and ASIC platforms.

5.1.3. Human-in-the-loop

Human-in-the-loop is a prominent approach in CPS, particularly in areas where human decision-making, oversight, or intervention is essential. By integrating humans into the control loop, this approach enables real-time interaction, supervision, and system adjustments driven by human input. While challenging to implement, advancements in digital technologies have greatly facilitated this integration. Studies [110–112] emphasize the importance of human involvement in the control loop, showcasing its benefits in real-time system interaction and adaptability. This methodology is crucial for various manufacturing applications, such as assembly tasks, quality control, decision-making support, and health risk assessments, ensuring enhanced safety, flexibility, and operational efficiency. Moreover, the human-in-the-loop paradigm extends to other fields like decentralized traffic merging and highway lane merging systems [113], where it significantly improves system performance and safety outcomes.

5.1.4. Standardized Benchmarks

CPS applications require precise synchronisation and robust optimisation to function effectively. However, developing standardized benchmarks for evaluating and comparing CPS solutions poses significant challenges. Below is a discussion of the key challenges and their consequences.

- Diversity in Application Requirements: CPS applications have highly varied requirements in terms of latency, fault tolerance, and real-time responsiveness. For example, autonomous driving systems require low latency and strict real-time synchronisation [24], whereas construction operations prioritize robustness and fault tolerance [9]. These differences make it difficult to create universal benchmarks that address the needs of all domains effectively.
- Heterogeneous Architectures: CPS systems involve a complex mix of hardware, software, and communication protocols. Variability in processing speeds, sensor accuracies, and network latencies requires synchronisation and optimisation solutions customized to diverse architectures. Standard benchmarks often fail to account for these architectural disparities.
- Dynamic Operating Environments: CPS must perform reliably in environments with unpredictable changes, such as varying workloads, communication delays, and environmental disturbances. Creating benchmarks that accurately simulate such dynamic conditions is a complex and resource-intensive task that makes standardisation challenging.

The following implications will be produced.

- Inconsistent Performance Metrics: Without common benchmarks, researchers and practitioners rely on ad hoc evaluation methods. This inconsistency makes it challenging to compare the efficiency, scalability, and effectiveness of different synchronisation and optimisation techniques.
- Limited Reproducibility: The absence of standardized frameworks impedes reproducibility, as the experimental setup and evaluation criteria vary widely between studies. This inconsistency hinders progress in developing reliable CPS solutions.
- Barriers to Collaboration: Standardized benchmarks foster collaboration by providing a shared foundation for evaluating CPS technologies. Without them, it becomes difficult for researchers, engineers, and domain experts to collaborate effectively within a cohesive ecosystem.
- Challenges in Real-World Applications: Many CPS applications, such as automotive systems and smart grids, require rigorous testing and validation to meet safety and performance standards. The lack of standardized benchmarks hampers this process, potentially affecting system reliability and trustworthiness.

Addressing the challenges outlined above would enable consistent performance evaluation, promote reproducibility, and encourage collaboration between disciplines. In addition, establishing robust benchmarks would improve the reliability and safety of CPS in real-world applications, contributing to the development of reliable and efficient systems.

5.1.5. Ethical and Societal Implications

The integration of ML-driven CV solutions into CPS presents a range of ethical and societal challenges that extend beyond technical performance. These challenges encompass legal, ethical, and social dimensions.

- Privacy Infringement
  - Implication: ML-based CV systems often process images and video data that can identify individuals in both public and private spaces. These images are frequently collected without explicit consent, raising ethical concerns regarding the unauthorized use of personal data. Even when datasets are anonymized, there remains a significant risk of re-identification through

advanced analytical techniques, particularly when combined with other data sources [?]. This presents substantial threats to individual privacy [?].

  - Mitigation: To address these concerns, policies and implementation practices must prioritize obtaining explicit consent from individuals before data collection, especially in sensitive environments. In cases where direct consent is impractical, visible signage, opt-out mechanisms, or public notifications should be provided to ensure transparency. Developers and system operators should adopt strong privacy frameworks grounded in informed consent and apply advanced anonymisation techniques when consent is not feasible. These anonymisation methods must be regularly updated to keep pace with emerging re-identification techniques [?]. In addition, encryption-based approaches offer further protection. Homomorphic Encryption allows computations on encrypted data without the need for decryption, while Secure Multiparty Computation (SMC) enables multiple users to collaboratively compute a function without revealing their data [114]. Finally, privacy-by-design principles should be integrated throughout the system's lifecycle, ensuring minimal data collection, transparency in data usage, and rigorous safeguards against misuse.

- Algorithmic Bias and Discrimination
  - Implications: CV systems trained on biased datasets can result in unfair outcomes such as misclassification or discriminatory treatment based on race, gender, or age. These datasets often encode societal biases, which are inadvertently learned and propagated by machine learning models. As a result, marginalized communities may face disproportionate harm, especially when these technologies are used in surveillance applications that operate without individuals' knowledge or consent, infringing on civil liberties. Furthermore, CV models may detect statistically valid but non-causal patterns, leading to incorrect or misleading conclusions. The effectiveness and fairness of these systems are heavily dependent on the quality and integrity of their training data. When the data is incomplete, outdated, or unrepresentative, the resulting models will inevitably reflect and amplify these deficiencies [115].
  - Mitigation: Addressing these challenges requires a multi-pronged approach. First, datasets should be carefully curated to ensure diversity and adequate representation of all demographic groups. Second, routine algorithmic audits using fairness metrics must be conducted to monitor and mitigate biased outcomes. Third, embedding explainability into the model design is essential to promote transparency, enabling stakeholders to interpret decisions and identify potential bias [114]. Moreover, involving ethicists, domain experts, and representatives from affected communities throughout the development lifecycle ensures that ethical considerations are prioritized and embedded from the outset [115].
- Safety and Adversarial Vulnerabilities
  - Implications: ML-driven CV systems are highly sensitive to data distribution shifts, adversarial inputs, and environmental variability, which can lead to unsafe or unintended decisions [116]. Overreliance on automated CV systems risks reducing essential human oversight, potentially leading to unchecked system behaviour in critical scenarios. These systems are also vulnerable to spoofing, tampering, and adversarial attacks, such as adversarial examples crafted to deceive neural networks into misclassifying objects, posing severe security risks [?]. Furthermore, the growing use of generative technologies capable of producing hyper-realistic yet fabricated audiovisual content introduces serious threats to democratic processes, including election manipulation and the erosion of public trust.
  - Mitigation: Enhancing the resilience of ML-driven CV systems requires integrating adversarial training and developing robust detection mechanisms to identify and neutralize spoofing or tampering attempts. Critical applications should maintain meaningful human-in-the-loop oversight to safeguard against over-automation and preserve accountability. Additionally, ethical system design must include built-in fail-safes and strong authentication protocols to

prevent the malicious use of synthetic media, such as deepfakes, thereby protecting both individual rights and societal stability.

- Labour Displacement and Job Restructuring
  - Implications: The integration of automated CV technologies into CPS (such as in retail checkout, quality control, and industrial inspection) poses a significant risk of displacing both manual and cognitive labour. While early automation concerns centred on physical jobs, generative AI now threatens knowledge-based professions across all educational levels. Increasingly, robots are performing roles in security, food service, and logistics, while AI is capable of completing tasks historically assigned to radiologists, software developers, designers, and journalists. Although automation can enhance efficiency and productivity, it also risks driving down wages and reducing employment opportunities, especially in the absence of adequate retraining infrastructure. The lack of robust vocational training and career transition programs further compounds the vulnerability of the workforce [**?**]. Nevertheless, some forecasts remain optimistic, suggesting that, with appropriate policy and educational reforms, AI and robotics could ultimately generate more jobs than they displace [117]. The central challenge lies in preparing workers to thrive in these emerging roles through strategic investments in education and human-centred skill development.
  - Mitigation: Addressing labour displacement requires proactive and collaborative action from policymakers, educators, and industry leaders. Large-scale reskilling and upskilling initiatives must be prioritized, emphasising digital literacy, adaptive learning, and human-AI collaboration. These programs should be tailored to prepare workers for evolving roles and enable lifelong learning pathways. In parallel, transitional support systems including career counselling, job placement services, and mental health resources can ease the socioeconomic impact on displaced individuals. Incorporating inclusive design principles that involve workers in co-developing AI systems can ensure that automation complements rather than replaces human labour [117]. Moreover, governments should create regulatory frameworks and incentive structures that encourage ethically aligned innovation, rewarding companies that pursue socially responsible automation. Collectively, these strategies foster an ecosystem where technological progress supports equity, resilience, and human dignity [**? ?**].

5.2. Interdisciplinary Perspectives

Cyberspace technology has seamlessly integrated into our modern world, underscoring the transformative synergy between ML, CV, and CPS. This interplay emphasizes the critical role of interdisciplinary collaboration in addressing complex challenges and driving technological innovation. Collaborative efforts among computer scientists, engineers, and domain experts are essential to harness the full potential of these technologies. The key points of this collaboration include:

5.2.1. Interoperability, complexity, and sustainability of CPS

- Interoperability: It refers to the ability of diverse systems or components to work together seamlessly. The development of CPS necessitates collaboration across multiple engineering disciplines, each specialising in different aspects of the product. Engineers, developers, and designers operate within various departments, each with distinct priorities and experience levels, relying on specialized software tools throughout the product life cycle [118]. To ensure proper system functionality and seamless integration of interconnected components, [119] introduces a domain-specific language for CPS, improved design, verification, and deployment of these intricate systems [120,121].
- Complexity: It arises from multiple interconnected components and variables, leading to unpredictability, non-linearity, and randomness. Addressing complexity requires interdisciplinary insights. [122] introduces the parallel systems method, based on Artificial systems, Computational experiments, and Parallel execution, to develop parallel intelligence, a cycle that iteratively

generates data, acquires knowledge, and refines real-world systems. This approach provides versatile solutions for managing complex systems across diverse fields.

- Sustainability: It refers to a system that could exist forever by balancing environmental, economic, and social factors, necessitating interdisciplinary strategies. [118] proposes integrating three mindsets (systems mindset, futuristic mindset, and design mindset) to navigate the uncertainty and complexity inherent in sustainability challenges.

5.2.2. Designing Effective Solutions

ML and CV technologies must be tailored to meet the objectives and constraints of CPS applications. While computer scientists design algorithms for tasks like object detection, engineers are tasked with integrating these algorithms into physical systems capable of real-time responsiveness. Domain experts ensure the system adheres to specific industry standards. For instance, in Wu et al.'s study [8], computer scientists develop algorithms for weak defect detection, engineers deploy these algorithms into production lines that operate in real-time, and domain experts validate the system's compliance with industry requirements and standards. Collaborative efforts are essential to create effective solutions that address both technical and domain-specific challenges.

5.2.3. Data Interpretation in Real-World Implementation

Interpreting data in CPS is complex due to the heterogeneous nature of the system, which involves sensors, embedded systems, AI algorithms, and human interactions. For example, in assembly operations [123], raw data collected from sensors such as cameras or LIDAR is structured and requires effective processing and interpretation using CV and ML techniques. Computer scientists and engineers focus on developing algorithms and system architectures while manufacturing experts ensure that the data are interpreted within the specific context of the real-world application to ensure the system responds appropriately, achieving outcomes like safety, performance, and efficiency. In addition, the operator plays a critical role by highlighting posture and the material handling process to provide unstructured feedback signals to the system for continuous adaptation and improvement [124,125].

5.2.4. Real-Time Decision-Making

In CPS, particularly in applications like autonomous driving or robotics, real-time decision-making is essential. ML algorithms created by computer scientists for perception and decision-making must work seamlessly with virtual leader systems to analyze sensor data. In the article by Yedilkhan et al [126], ML models improve obstacle avoidance strategies through learned behaviours from prior data to handle uncertainty and adapt to dynamic environments. Engineers ensure that these systems are optimized for real-time performance and reliability. Collaboration with domain experts ensures that the systems are not only accurate, but also safe, efficient, and compliant with industry standards.

The development of ML and CV systems for CPS is an ongoing process that requires constant feedback. Domain experts can provide valuable insights from real-world testing, helping engineers and computer scientists fine-tune algorithms. Collaboration enables continuous improvement by ensuring that the system is iteratively refined to address new challenges and incorporate emerging technologies [127,128].

5.3. Emerging Trends

5.3.1. Edge Artificial Intelligence

Edge Artificial Intelligence (AI) is a groundbreaking computing paradigm designed to perform machine learning model training and inference directly at the network edge [129]. This paradigm enables two distinct approaches [55]: AI on edge, where models are trained and inferred either collaboratively through direct interaction between edge devices or using local edge servers near these devices, and AI for edge, which focuses on integrating artificial intelligence into edge computing architectures. This integration enhances edge devices' ability to handle complex data processing and decision-making tasks. Although relatively new, the field has experienced remarkable growth recently, driving innovative CPS applications[130,131].

- Real-Time Processing and Low Latency: Edge AI revolutionizes real-time decision-making processes by enabling on-device data processing, which minimises latency and ensures instant responses. This capability is indispensable for applications that demand immediate and reliable decision-making, such as autonomous vehicles and health care. In these scenarios, rapid responses are not only beneficial but also critical. For example, automotive vehicle systems require handling vast amounts of heterogeneous data from various sensors, requiring high-performance and energy-efficient hardware systems to process this information in real-time, interacting between functional modules seamlessly with low overhead, and facing strict energy constraints, emphasising the need for optimised hardware and computational techniques. By decentralising intelligence, edge AI brings ML model training and inference directly to the network, enabling communication between edge systems and infrastructure, and reducing the computational burden on the edge systems [10].

  Edge AI is a transformative technology that brings numerous benefits to the functionality and efficiency of medical devices, especially in the realm of the Internet of Medical Things (IoMT) [132]. By processing data locally, Edge AI ensures faster, real-time decision-making, crucial in medical contexts. For instance, in remote monitoring systems, critical health alerts can be instantly generated and communicated to caregivers or medical professionals, improving the reliability and responsiveness of these systems. In such cases, local storage capacities and synchronisation of sensor data may cause challenges to the application creators.
- Enhanced Security and Privacy: Edge AI minimises the need to transmit sensitive data to central servers, significantly enhancing the security and privacy of decentralized CPS applications. This localized processing not only reduces exposure to potential data breaches but also strengthens the overall resilience of the system. Ensuring the reliability, security, privacy, and ethical integrity of edge AI applications is paramount, as edge devices handle sensitive information with potentially severe consequences in the event of a breach. Robust encryption methods, stringent access controls, and secure processing and storage frameworks are indispensable for safeguarding data and maintaining trust [129]. Hardware-supported Trusted Execution Environments are often employed to enhance security by isolating sensitive computations. However, these solutions present challenges related to performance and integration, necessitating a delicate balance between maintaining robust security and ensuring efficient system operations. Addressing these challenges is critical for the successful deployment of edge AI in secure and decentralized CPS environments.
- Energy Efficiency: The growing demand for AI applications highlights the need for energy-efficient and sustainable edge AI algorithms. Advanced AI, particularly deep learning, consumes substantial energy, posing sustainability challenges. Developing lightweight and energy-efficient AI models is essential for supporting edge devices with limited computational resources, thereby enhancing the sustainability of CPS applications. Computational offloading is another effective method to reduce energy consumption in edge devices [132].

  However, achieving a balance between high performance and energy efficiency is crucial. Often, small gains in accuracy require significantly more energy, which is inefficient and environmentally unsustainable when ultrahigh accuracy is not necessary. Researchers must carefully evaluate the trade-offs between accuracy and energy use.

  For the significant impact of energy consumption during the operation, production, and lifecycle of edge devices, creating durable, upgradeable, and recyclable devices is vital to minimise ecological impact. Implementing policies to promote energy-efficient AI and regulating the environmental footprint of device manufacturing and disposal are critical steps toward achieving sustainability in edge AI [129].
- Interoperability: Efforts are being made to develop comprehensive standards and frameworks to ensure seamless interoperability between edge devices and CPS components across diverse

applications. These standards aim to establish uniform protocols for data exchange, device communication, and system integration, enabling heterogeneous edge devices and CPS components to work together cohesively. This interoperability is critical for supporting scalability, reducing system fragmentation, and fostering a more unified ecosystem that can accommodate advancements in hardware and software technologies.

Moreover, the development of such frameworks addresses challenges related to compatibility, security, and system resilience, providing a robust foundation for reliable decentralized operations. These initiatives also incorporate mechanisms to manage dynamic environments, where edge devices and CPS components must adapt to changing conditions in real time while maintaining performance and reliability [133].

5.3.2. Self-adaptive Systems Leveraging Reinforcement Learning (RL)

Self-adaptive systems are pivotal in addressing the dynamic and uncertain demands of modern technology landscapes. These systems adjust their behaviour autonomously to maintain optimal performance despite changes in their environment or internal state. While traditional approaches to adaptation rely on predefined rules or models created during design time, these methods struggle to cope with the unpredictable and complex nature of real-world environments. Reinforcement learning (RL) has emerged as a transformative solution, empowering self-adaptive systems with the ability to learn, adapt, and optimise decisions dynamically.

- Addressing Design-Time Uncertainty: One of the most significant challenges in developing self-adaptive systems is the uncertainty inherent at design time. Online RL provides a compelling solution [134]. By enabling systems to learn directly from interaction with their environment, RL equips self-adaptive systems with the ability to respond effectively to previously unencountered conditions. This adaptive capacity is critical for systems deployed in dynamic environments, such as autonomous vehicles or distributed cloud-edge networks, where operational contexts can shift unpredictably.
- Real-Time Decision-Making: The ability to make real-time decisions is the cornerstone of self-adaptive systems. RL excels in this domain by continually refining its policies based on operational feedback, ensuring the system remains responsive to changes. RL-driven systems autonomously optimize their behaviour, balancing competing objectives such as performance, energy efficiency, and reliability [134]. This capability is particularly valuable in applications like IoT-driven healthcare, where immediate responses to patient data can be life-saving, and in autonomous systems, where split-second decisions are vital for safety.
- Enhancing Efficiency: Efficiency is a critical consideration in the operation of self-adaptive systems. RL supports this by enabling dynamic resource allocation and optimising the use of computational, energy, and network resources based on current demands. Deep RL integrates energy optimisation with load balancing strategies, in order to minimise energy consumption while ensuring server load balance under stringent latency constraints [135]. Furthermore, RL's ability to handle nonlinear and stochastic environments makes it particularly well-suited for real-world applications, where unpredictability and instability are the norm. This adaptability ensures robust performance in dynamic and challenging conditions, reinforcing its utility across various domains.
- Generalisation and Scalability: Deep RL extends the capabilities of RL by integrating neural networks to represent the learned knowledge. This allows self-adaptive systems to generalize their learning to unseen states and handle high-dimensional input spaces, such as sensor data or video streams. This generalisation capability is crucial for scalability, enabling RL-driven self-adaptive systems to operate effectively in diverse and complex environments. Applications such as smart cities, where systems must manage vast amounts of real-time data from interconnected devices, benefit immensely from Deep RL's scalability and adaptability.

5.3.3. Hybrid Machine Learning Models

The rapid advancements in machine learning have led to the emergence of hybrid models that combine DL with traditional algorithms to achieve improved efficiency, flexibility, and scalability in diverse applications. These hybrid approaches aim to harness the strengths of both paradigms while mitigating their respective limitations.

- Enhanced Performance: Deep learning excels at extracting high-level features from unstructured data, such as images and text. However, it often requires significant computational resources. Traditional algorithms handle structured data and provide clear interpretability [136]. In [137], authors applied CNN and autoencoders to extract features and then followed by the particle swarm optimisation (PSO) algorithm to select optimal features and reduce dataset dimensionality while maintaining performance. Finally, the selected features were classified by the third stage using learnable classifiers decision tree, SVM, KNN, ensemble, Naive Bayes, and discriminant classifiers to process the acquired features to assess the model's correctness. Combining these techniques results in models that deliver high performance without the prohibitive costs of standalone deep learning methods.
- Improved Generalisation: Hybrid models combine the strengths of deep learning and traditional algorithms, capitalising on deep learning's ability to handle complex, non-linear relationships in data while utilising traditional methods to enhance interpretability and generalisation, particularly in scenarios involving smaller datasets. For example, the Adaptive Neuro-Fuzzy Inference System (ANFIS), as discussed in [137], exemplifies a hybrid network where fuzzy logic intuitively models nonlinear systems based on expert knowledge or data. Neural networks complement this by introducing adaptive learning capabilities, enabling the system to optimise parameters such as membership functions through input-output data. This integration empowers ANFIS to effectively model complex, nonlinear relationships, making it highly applicable in tasks such as prediction, control, and pattern recognition.
- Scalability and Adaptability to Diverse Tasks: Hybrid models offer remarkable flexibility, enabling customisation for specific applications by integrating the most advantageous features of distinct paradigms. In [138], by combining Statistical Machine Translation (SMT), which uses statistical models to derive translation patterns from bilingual corpora, with Neural Machine Translation (NMT), which employs Sequence-to-Sequence (Seq2Seq) models with RNNs and dynamic attention mechanisms, these approaches capitalize on the statistical precision of SMT and the contextual richness of neural networks. Additionally, ensemble methods enhance translation quality further by amalgamating multiple models, proving particularly effective for domain-specific adaptations and ensuring robust performance.
- Limitations: Hybrid learning systems offer robust solutions for complex data-driven challenges by combining the strengths of both methodologies. However, they face several challenges [136], including high model complexity, which complicates configuration, optimisation, and interpretation. Despite advances in transparency, their layered architecture often obscures decision-making processes, raising issues of interpretability. The extensive and diverse datasets required for training pose significant privacy and security risks. Additionally, deploying and maintaining these systems is resource-intensive due to their sophisticated architecture and the need for regular updates to stay aligned with evolving data and technologies. Real-time processing capabilities can be hindered by the computational intensity of DL components, and the energy demands of training and operating hybrid models raise environmental concerns. Long-term maintenance further demands substantial effort to ensure these models remain effective and relevant in dynamic environments.
- Future Research: Future research in hybrid learning should focus on deeper interdisciplinary integration with fields like cognitive science, medicine, and computing to achieve AI systems that more closely emulate human cognition. Advancing model generalisation is equally critical, emphasising the development of adaptive systems capable of autonomously adjusting to varying datasets and environmental conditions. Additionally, enhancing AI accessibility is essential

to democratize its use, and improve educational resources, and community-driven initiatives, thereby broadening the impact of AI as a universal problem-solving tool [136].

5.3.4. Vision Foundation Models

Foundation models are large-scale, general-purpose AI systems trained on vast amounts of data, generally using self-supervision, to learn generalizable patterns across modalities. In NLP, examples of LLMs like BERT, GPT-3, GPT-4, and MPT-30B, have enabled the development of conversational agents and language understanding systems tailored to specific tasks. [139,140] Visual foundation models (VFMs) extend this concept to the visual domain, encompassing systems trained to understand and reason about visual data, such as images and videos, often alongside other modalities (text or audio). These models can perform a wide range of tasks, including object detection, segmentation, image captioning, and visual question answering (VQA) [141]. A defining characteristic of VFMs is their promptability: they can be adapted to new tasks using simple inputs, such as textual prompts or visual cues, eliminating the need for extensive retraining. Their flexible and multi-modal architecture makes them a key technology for building intelligent, human-aligned systems. However, challenges remain in areas such as bias mitigation, interpretability, efficiency, and generalisation to real-world environments.

- Segment Anything Model (SAM): A foundational vision model is designed for promptable image segmentation tasks. Trained in the large-scale SA-1B dataset, comprising over 1 billion segmentation masks in 11 million images, SAM demonstrates strong zero-shot performance, generalising to new image distributions and tasks without task-specific fine-tuning [142]. SAM has become a cornerstone for high-level vision tasks such as image segmentation, captioning, and editing. However, its high computational cost due to its transformer-based architecture limits its practicality in real-time or resource-constrained industrial settings.
  To address these limitations, [143] proposed FastSAM, a real-time, CNN-based alternative. By reframing segmentation as an instance segmentation task with prompting and training on only 1/50 of the SA-1B dataset, FastSAM achieves performance comparable to SAM while operating at 50× faster inference speed, making it more suitable for real-time and edge deployments.
  Building on the SAM foundation, [144] reviewed the SAM family, highlighting SAM2's enhancements in segmentation accuracy and a streaming memory architecture that enables real-time video segmentation. In contrast, [145] proposed the Fovea-Like Input Patching (FLIP) approach, which encodes visual input in an object-centric, off-grid manner from the outset. By decoupling spatial location encoding from perceptual object features, FLIP enhances data efficiency and excels at segmenting small objects within high-resolution scenes. It achieves Intersection-over-Union (IoU) scores comparable to SAM with lower computational overhead and consistently outperforms FastSAM across benchmarks such as Hypersim, KITTI-360, and OpenImages.
- Multimodal Large Language Models (MLLMs): These are advanced AI systems capable of processing and generating across multiple modalities, typically text and images, and probably, video and audio. These models support a wide range of tasks such as image captioning, VQA, and even text-to-image generation. At their core, MLLMs integrate visual encoders with language models like GPT or BERT to enable reasoning over both modalities. A prominent example of a visual encoder is CLIP (Contrastive Language–Image Pretraining). CLIP learns joint image-text representations through contrastive training on 400 million (image, text) pairs from the internet, using dual encoders, one for images and one for text. This enables strong zero-shot generalisation to novel image domains via text prompts [141]. While CLIP's language grounding and cross-modal capabilities are powerful, its coarse-grained representations and lack of pixel-level detail limit its effectiveness for dense prediction tasks like semantic segmentation.
  Self-distillation with No Labels (DINO) is a vision-only, self-supervised model that learns high-quality visual features without requiring labelled data. It uses a self-distillation strategy, where a student ViT is trained to match the output of a momentum-updated teacher network. DINO excels at discovering semantic object boundaries and performs well in tasks such as unsupervised

object segmentation and fine-grained categorisation [146]. Unlike CLIP, DINO is not trained with text, making it unsuitable for prompt-based or language-guided tasks. However, recent advances with DINOv2 demonstrate that with a large and diverse curated image dataset and improved training stability, vision-only models can produce universal features that outperform CLIP on many image-level and pixel-level benchmarks [147].

Further insights into DINOv2's architecture reveal that its different layers capture complementary information: lower layers specialize in fine-grained details useful for localisation, while deeper layers encode more global semantic concepts [148]. This insight has led to the design of multi-level feature fusion strategies. Despite being unimodal, DINOv2's rich spatial representations can be effectively aligned with language models using a simple multilayer perception alignment head, making it competitive within MLLM architectures. In contrast, other visual backbones such as Masked Autoencoder (MAE) lack semantic richness, and Data-efficient Image Transformer (DeiT) models, being strongly supervised, struggle with cross-modal alignment [148].

To bridge the gap, a hybrid vision encoder (CLIP + DINOv2) (COMM) leverages the global-semantic strengths of CLIP and the fine-grained, structure-aware features of DINOv2, resulting in superior performance across a range of MLLM benchmarks. This hybrid approach exemplifies the trend toward combining vision foundation models for robust multimodal understanding [148].

5.4. Research Gaps

The integration of ML techniques, particularly in CV, within CPS, has unlocked significant advancements in fields such as autonomous vehicles, smart cities, and industrial automation. These systems rely heavily on synchronisation methods to ensure that distributed components collaborate effectively. However, the scalability of these synchronisation methods remains a critical challenge, compounded by insufficient attention to real-world deployment issues and the lack of adaptive models for handling diverse and dynamic CPS environments. This subsection explores the key research gaps that hinder progress in this domain and highlights directions for future work. Identifying these research gaps is critical to improving the development of an efficient and resilient CPS. Some vital areas are outlined below.

5.4.1. Limited Scalability of Synchronisation Methods

- Resource constraints in CPS: One of the most prominent issues with current synchronisation methods is their limited scalability in CPS environments. These systems often operate under resource-constrained conditions, with devices such as sensors, cameras, and actuators constrained by bandwidth, energy, and computational power. Many existing synchronisation techniques assume abundant resources, which is unrealistic in practical CPS deployments. We expect to have methods to balance the accuracy of synchronisation with the energy efficiency and computational cost. Consequently, there is a need for lightweight synchronisation algorithms that optimize resource usage without compromising accuracy or efficiency.

  In [149], the results show an improvement in precision with an increasing sampling rate at the cost of increased memory consumption and computation time. Similarly, in [7], post-deployment processing to align and synchronize data streams introduces computational overhead, which can challenge resource-constrained systems with limited processing power or memory. Moreover, in [**?**], the synchronisation approach is based on standard components, which may have limitations in terms of precision and robustness, especially when scaling up or requiring higher performance.
- Bottlenecks in distributed systems: Distributed systems, another core aspect of CPS, face significant bottlenecks in synchronisation due to the communication overhead and latency associated with global updates. This is particularly problematic in real-time applications like autonomous vehicles, where even minor delays can have critical consequences. One possible solution shown in [150] is the use of a polychronous model of computation for concurrent systems to free programming from synchronous timing models and to enhance robustness against clock synchronisation

failure-based attacks. This approach allows processes to execute and communicate at their paces without requiring rigid synchronisation, thereby reducing bottlenecks caused by contention for shared resources.

However, current research has not sufficiently addressed techniques to minimise communication requirements, such as using model pruning, gradient sparsification, or local aggregation. Becker et al. [151] show that contention among system modules severely affects latency and performance predictability, but LiDAR-related components contribute significantly to system latency and even high-end CPU and GPU platforms cannot achieve real-time performance for the complete end-to-end system. Furthermore, ensuring a balance between local computation and global model updates remains an unresolved challenge. Hybrid synchronisation techniques that adapt dynamically to the system's real-time state could address this issue, but their development is still in its infancy.

5.4.2. Insufficient Focus on Real-World Deployment Challenges

- Environmental Variability: The real-world deployment of ML-based synchronisation methods in CPS introduces a host of challenges that have not received sufficient attention. One major issue is the variability of real-world environments, which often include unpredictable network latency, device failures, and dynamic workloads. Current synchronisation methods are not robust enough to handle these variations, and research on fault-tolerant approaches that can recover gracefully from such disruptions is limited. Developing methods that maintain performance despite environmental variability is essential to advance the reliability of CPS.
- Deployment at scale: Many synchronisation methods are tested in controlled environments or small-scale settings, which do not reflect the challenges of real-world CPS deployments involving hundreds or thousands of nodes. In [102], the datasets and scenarios used are relatively simple and may not fully demonstrate the generality of the proposed cognitive Digital Twin architecture. More experiments in real industrial maintenance scenarios are needed to validate performance in multi-task, resource-allocation contexts involving personnel, spare parts, and materials. Additionally, the example focuses on updating microservices and the knowledge graph post-data analytics, rather than the physical-world operational responses. Future iterations could incorporate deviations between expected and actual outcomes into the self-evolution process to develop more realistic maintenance solutions.
- Real-time constraints: Real-time constraints further complicate deployment. Many CPS applications, such as surveillance and industrial automation, require synchronisation methods that can operate in real time to process high-frequency data streams. However, the latency introduced by current synchronisation methods makes them unsuitable for such applications. Research on event-driven or asynchronous synchronisation mechanisms that prioritize low-latency processing is still nascent and demands further exploration.

5.4.3. Adaptive Models for Diverse CPS Environments

- Heterogeneity in devices: The diversity of CPS environments presents another significant research gap. These systems often involve a wide range of devices with varying capabilities, such as sensors, drones, and cameras, each with different levels of processing power, storage, and communication bandwidth. Current synchronisation methods are not designed to account for this heterogeneity, leading to inefficiencies in resource utilisation. Adaptive synchronisation algorithms that dynamically adjust to the capabilities and constraints of individual nodes are needed to address this gap.
- Diverse tasks: CPS tasks vary widely, ranging from object detection to anomaly detection and action recognition. Each task has unique synchronisation requirements, but current methods often adopt a one-size-fits-all approach, failing to optimize for the priorities of individual tasks. Developing task-specific synchronisation strategies and exploring multi-tasking synchronisation approaches could significantly enhance the performance and flexibility of CPS.

- Dynamic environments: Dynamic environments pose further challenges, as CPS systems often operate under non-stationary conditions where data distributions, network topologies, or operational requirements can change over time. Existing models lack the adaptability to handle such conditions effectively. Self-learning synchronisation methods that adjust based on feedback from the environment offer a promising direction for future research, enabling models to remain robust and effective in evolving CPS scenarios.

5.4.4. Integration with Emerging Technologies

- Edge and Federated Learning: Edge and federated learning paradigms have shifted the focus from centralized to decentralized systems, necessitating new synchronisation strategies tailored for these frameworks. Efficient edge-to-cloud synchronisation techniques and privacy-preserving methods for federated learning are critical areas that require further investigation.
- Neuromorphic Computing and Event-Based Vision: CNNs have achieved remarkable success in computer vision but remain computationally expensive and energy-intensive, making them less practical for edge devices and real-time applications. In contrast, neuromorphic computing, inspired by brain neural processes, offers a promising alternative by leveraging spiking neural networks (SNN) and event-driven processing to create efficient and biologically plausible visual recognition systems. However, despite its advantages in power efficiency and real-time processing, neuromorphic computing is still primarily a research tool rather than a mainstream solution. Most current applications focus on benchmark datasets, and neuromorphic systems have yet to outperform deep learning models in accuracy consistently [152].

  However, [53] highlighted a potential misconception where SNNs may seem more energy-efficient than ANNs truly are.. A significant limitation of neuromorphic computers is their heavy dependence on traditional host machines for defining software structures and managing communication with sensors and actuators. This reliance can reduce the performance benefits of neuromorphic computing, as the overhead costs of data transfer and processing on conventional systems may offset its efficiency advantages. A key challenge for the future is to minimise the dependence on conventional computers and optimize communication architectures to fully unlock the potential of neuromorphic computing [152,153].

  One of the most promising applications of neuromorphic computing in computer vision is edge computing. When integrated with event cameras, neuromorphic systems enable low-latency, real-time applications, making them ideal for high-speed object detection, motion tracking, semantic segmentation, optical flow estimation, and 3D vision in resource-constrained environments [154]. These capabilities are particularly valuable for autonomous systems, robotics, and surveillance, where real-time decision-making with minimal power consumption is crucial.

  Another key challenge is the integration of curiosity-based SNNs with traditional deep learning models or other machine learning paradigms. Hybrid approaches could combine the strengths of both deep learning and neuromorphic computing, but this integration requires significant research into training methodologies, architectural compatibility, and hardware support. Developing efficient hybrid neuromorphic-deep learning frameworks could unlock new capabilities in adaptive vision systems, self-learning models, and real-time AI applications [152].

6. Recommendations

6.1. For Practitioners

This study explores strategies and techniques to synchronize, optimise, and adapt ML models within CPS. Despite their potential, deploying ML models in CPS presents unique challenges, including the demands for real-time processing, security, and reliability across varied physical environments. Successful deployment requires leveraging advanced methodologies such as edge computing to enable low-latency applications and federated learning to ensure secure, distributed data processing. When combined with other practical implementation tasks, these approaches can substantially improve the robustness and efficiency of CV systems in CPS.

6.1.1. Implementing Edge Computing

Edge computing plays a critical role in enabling real-time decision-making for CPS applications such as autonomous vehicles, industrial automation, and surveillance systems. By processing data locally on edge devices instead of relying on centralized cloud servers, edge computing significantly reduces latency and enhances responsiveness. This approach is vital for high-stakes applications where low latency is essential, and it also minimises bandwidth usage and reduces reliance on stable internet connectivity. The following key tasks are fundamental to a successful implementation of edge computing in CPS:

- Model optimisation: Apply techniques [13,25,48] such as pruning, quantisation, and knowledge distillation to develop lightweight models that operate effectively in resource-constrained edge devices without sacrificing accuracy.
- Hardware Utilisation: Select hardware platforms designed specifically for edge computing, such as NVIDIA Jetson, Intel Movidius, or Google Coral. Additionally, leverage accelerators such as GPUs and TPUs to improve computational efficiency.
- Runtime Frameworks: optimised inference frameworks, including TensorRT, ONNX Runtime, or PyTorch Mobile, are used to ensure efficient and reliable execution of models on edge devices.
- Partitioning Workloads: Distribute tasks strategically between edge devices and the cloud. Execute time-sensitive computations on the edge while offloading resource-intensive processes, such as retraining or extensive analytics, to cloud infrastructure.
- Real-Time Monitoring: Develop mechanisms to continuously monitor the performance and health of edge devices, ensuring reliability and consistent operation even in varying environmental conditions.

6.1.2. Leveraging Federated Learning

FL is a decentralized method of training machine learning models that enables edge devices to collaborate without requiring the sharing of raw data. This approach is particularly advantageous in CPS where data privacy and security are critical, such as in healthcare, smart cities, and defence applications. FL not only enhances data security by keeping sensitive information local but also facilitates the creation of models that are representative of diverse environments, thereby improving generalisation and reducing bias. Key tasks for implementing FL effectively include the following:

- Data Locality: Maintain sensitive data on local devices and transfer only model updates to a central server. This approach minimises the risk of data breaches and ensures compliance with privacy regulations such as GDPR.
- Communication Efficiency: Employ techniques like model compression, update sparsification, and asynchronous communication to reduce the bandwidth required for transmitting model updates between devices and the central server.
- Security Measures: Implement safeguards such as differential privacy and secure multi-party computation to protect data and model parameters from adversarial attacks and ensure the integrity of the learning process.
- Federated Averaging: Utilize aggregation algorithms, like FedAvg, to effectively combine model updates from multiple devices and enhance the robustness of the aggregation process by integrating outlier detection or Byzantine-resilient methods to handle potentially malicious updates.
- Heterogeneity Handling: Design systems capable of accommodating a wide range of edge devices with varying computational capabilities and network conditions. This can be achieved by dynamically allocating tasks based on each device's capabilities, ensuring efficient and equitable participation in the learning process.

6.1.3. Integration of Edge Computing and Federated Learning with CPS

Seamless integration of edge computing and federated learning with CPS is essential to ensure harmonious operation between machine learning (ML) models and the system's physical components. This integration supports real-time decision-making, sensor fusion, fault tolerance, and scalability. We discuss key factors influencing effective integration below:

- Monitoring and Continuous Learning: To remain effective in dynamic environments, edge computing and federated learning systems require ongoing monitoring and adaptability. Key elements include:
  - Performance Metrics Tracking: Continuously monitor model performance metrics, such as latency, accuracy, and confidence, to promptly detect and address potential issues.
  - Periodic Updates and Retraining: Incorporate mechanisms for regular model updates and retraining with newly collected data to maintain accuracy and relevance.
  - Anomaly Detection: Implement systems to flag anomalous data or behaviour, enabling swift intervention when deviations from expected operations occur.
- Security and Privacy: Robust security and privacy measures are vital for safeguarding data and ensuring trust in CPS operations. These measures include:
  - End-to-End Encryption: Secure all data transmissions between edge devices, cloud servers, and central aggregation points using encryption.
  - Role-Based Access: Restrict access to models, data, and system components based on user roles and authentication protocols.
  - Defenses Against Adversarial Attacks: Employ strategies like input sanitisation, adversarial training, and anomaly detection to protect against malicious activities.
- Ethical and Regulatory Compliance: Ethical standards and regulatory requirements are critical for public trust and the lawful deployment of ML models in CPS. Key considerations include:
  - Transparency: Provide clear and transparent documentation and explanations of model operations, especially in safety-critical applications.
  - Standards Compliance: Ensure adherence to relevant standards and laws, for example, ISO 26262 for automotive safety [155] and GDPR for data protection [156], to guarantee ethical and legal deployment.

6.2. For Researchers

CPS frequently operates on edge devices with limited computational resources, memory, and energy capacity. This constraint is especially critical in domains like IoT devices, autonomous drones, and wearable technologies, where real-time decision-making and prolonged operation are essential. To address these challenges, lightweight ML models must be designed to effectively balance computational efficiency and accuracy.

6.2.1. Developing Lightweight ML Models for Resource-Constrained CPS

Approaches such as model compression, pruning, quantisation, and knowledge distillation (discussed in Section IV.B.2) are promising for reducing model size and complexity [**?**]. These techniques help optimise resource utilisation without significantly compromising model performance. Future research can delve deeper into these methodologies, emphasising the need to maintain interpretability and robustness in constrained environments.

A particularly promising tool for designing resource-efficient ML models is Neural Architecture Search (NAS). NAS automates the development of high-performing neural networks tailored to specific resource constraints, requiring minimal human intervention. Its framework comprises three core components [157,158]:

- Search Space (SSp): Defines the range of architectures that NAS can explore. Advances in SSp have broadened the scope of candidate designs, enabling the discovery of innovative architectures that were previously unattainable.
- Search Strategy (SSt): Encompasses methods for exploring the defined search space. Recent research has focused on improving the efficiency of search strategies, and optimising the balance between computational resources and performance outcomes.
- Validation Strategy (VSt): Refers to the techniques used to evaluate the performance of candidate architectures. Enhanced validation strategies have increased the reliability of NAS results while minimising the time and resources required for evaluation.

Although NAS is still in its early stages, it holds immense potential. Its applications are expected to extend beyond image classification into domains requiring complex network designs, such as multi-objective optimisation, model compression, and advanced tasks like object detection and semantic segmentation [155].

Moreover, the integration of NAS with emerging technologies like federated learning and edge computing presents exciting possibilities. These synergies could enable real-time, distributed model optimisation, fostering the development of scalable and adaptive ML systems. By leveraging such advancements, NAS can facilitate the creation of robust, efficient, and resource-aware ML models that are well-suited for dynamic CPS environments. [158].

In conclusion, developing lightweight ML models tailored for resource-constrained CPS is a critical research direction. By combining techniques like model optimisation with automated solutions such as NAS, the field can achieve breakthroughs in efficiency, scalability, and adaptability, paving the way for innovative CPS applications.

6.2.2. Exploring TL for Faster Adaptation to New Tasks

Transfer learning is a machine learning approach where knowledge acquired in solving a source task is applied to a related but different target task. By utilising pre-trained models or previously learned knowledge, transfer learning accelerates learning, reduces dependence on extensive labelled data in the target domain, and enhances performance, particularly in data-scarce or computation-constrained.

Transfer learning provides a promising solution in CPS, which often operates in dynamic environments or faces conditions different from their original training scenarios. By enabling models to adapt rapidly to new tasks or domains using pre-trained knowledge, TL minimises the need for large, annotated datasets and computationally intensive retraining. However, domain shift, where the source and target domains have different data distributions, remains a significant challenge [159]. For example, a model trained on clean, curated datasets might perform poorly on noisy, real-world data. Overcoming domain shift requires methods like domain adaptation to reduce the distributional differences between the source and target domains. [160] and [161] have made significant progress but exhibit certain limitations (no learning features adaptively from the source and using a discontinuous decision strategy). Recent techniques introduced by [162] enhance outcomes by identifying and leveraging transferable structures in high-dimensional settings, offering robust theoretical guarantees and empirical benefits. Other innovative approaches [159,163] include :

- Few-Shot Learning: Facilitates robust training with minimal data in the target domain.
- Zero-Shot Learning: Uses cross-domain knowledge to predict unseen target classes without any labelled data.
- Generalisation: Focuses on transferring knowledge across related tasks (e.g., object detection and semantic segmentation) to save retraining time and resources.
- Federated Transfer Learning (FTL): Extends TL to decentralized, privacy-sensitive contexts where source and target data cannot be shared or centralized [164].

6.3. *For Policy Suggestions*

Developing individual automotive CPS using best-effort technologies is feasible but fraught with challenges due to technical complexity, high costs, and the potential for errors. To address these issues and support CPS development, several policies and guidelines can be implemented:

6.3.1. Establish Standardized Benchmarks for ML Performance in CPS

Standardized benchmarks are essential for evaluating and comparing machine learning models used in CPS. They ensure consistent, reliable performance metrics aligned with real-world applications. However, no such repository exists for automotive CPS. The recommendations are below.

- Create comprehensive benchmark repositories to include typical and worst-case models.
- Mitigate IP and security concerns by forming regulatory bodies to redact and standardize real-world models.
- Develop industry-specific benchmarks tailored to the needs of various CPS domains.
- Mandate performance evaluations against benchmarks to ensure baseline performance and reliability prior to deployment.

6.3.2. Promote Open-Source Tools

Open-source tools are transformative, fostering innovation, accessibility, and collaboration while lowering costs. The key benefits are:

- Collaboration: Enable global developers to share ideas and improvements, driving innovation and creative problem-solving.
- Accessibility: Make CPS development affordable by eliminating proprietary software costs, levelling the playing field for all.
- Transparency: Build trust through open review, auditing, and enhancement of code, promoting ethical and reliable systems.
- Accelerated Development**: Leverage pre-trained models and ready-to-use tools to reduce development time significantly.

6.3.3. Encourage Cross-Domain Collaboration

CPS development spans diverse sectors, benefiting from shared knowledge and multidisciplinary expertise. The recommendations are:

- Foster research programs that integrate engineering, computer science, economics, and social sciences.
- Establish CPS innovation hubs to tackle common challenges like security, real-time data processing, and model generalisation.
- Create industry consortia to set shared goals, develop common standards, and address deployment challenges collaboratively.

6.3.4. Regulate and Promote Ethical Use of CPS

CPS interacts with physical processes and human lives, necessitating ethical deployment, privacy protection, and transparency. The recommendations are:

- Develop ethical guidelines and regulations to ensure CPS operates safely and fairly, especially in sensitive sectors, like healthcare and autonomous vehicles.
- Support explainable CPS for transparency in machine learning models, fostering public trust and accountability.
- Enforce robust cybersecurity and privacy standards to safeguard physical assets and data against attacks or misuse.
- Promote diversity in research and development teams to create inclusive CPS technologies that consider diverse societal impacts

doi:10.20944/preprints202501.0521.v3

7. Conclusion

7.1. Summary of Findings

7.1.1. CNN

CNNs are essential in CV applications within CPS due to their ability to automatically learn spatial features and patterns from images. CNNs excel in tasks like image classification and object detection through their layered architecture, which includes convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for decision-making. Parameter sharing and efficient visual data processing make CNNs foundational to modern CV applications. Recent advancements in CNN-based methods include:

- R-CNN: A two-stage object detector with high accuracy but computationally intensive. Variants like Fast R-CNN, Faster R-CNN, and Mask R-CNN improve efficiency and extend functionality to instance segmentation and pose estimation.
- ResNet: Focuses on training deep networks efficiently using residual learning and skip connections, widely applied in image classification, segmentation, and as a backbone for detection models.
- YOLO: A single-stage real-time detector that processes images in one pass, offering high speed with moderate localisation errors. Versions like YOLOv2–YOLOv5 enhance accuracy while maintaining real-time performance, making them suitable for applications like autonomous vehicles.
- SSD: Alternatives to YOLO that improve detection accuracy and address specific limitations in network design and loss functions.

7.1.2. Federated Learning

FL is a transformative approach for synchronising distributed CPS by enabling collaborative model training across multiple devices while keeping data localized. This decentralized methodology enhances privacy, reduces the need for centralized data storage, and supports continuous learning, making it ideal for sensitive, large-scale, and real-time applications like autonomous vehicles and industrial robotics. Key advantages of FL for CPS include:

- Privacy Preservation: Retains data on local devices, sharing only model updates to protect sensitive information.
- Scalability: Efficiently handles extensive distributed networks with numerous devices.
- Reduced Latency: Minimises communication overhead by processing data locally.
- Heterogeneity Handling: Adapts to diverse and resource-imbalanced environments.
- Robustness and Adaptability: Supports continuous learning and dynamic updates to maintain model reliability.

FL employs two synchronisation techniques:

- Synchronous FL: Updates are synchronised simultaneously, challenging in heterogeneous environments.
- Asynchronous FL: Updates occur independently, improving flexibility but risking stale updates.

Challenges include handling non-IID data across nodes, reducing communication overhead, and addressing computational disparities among devices. Despite these, FL's ability to facilitate decentralised collaboration, real-time adaptation, and privacy-conscious coordination makes it pivotal for CPS synchronisation and scalability in modern smart systems.

7.1.3. Meta-learning

Meta-learning in CV focuses on creating models capable of quickly adapting to new tasks with minimal data and computational resources. It is particularly beneficial for tasks with scarce data, enabling models to generalise across diverse domains and applications. Meta-learning techniques emphasise extracting broadly applicable features for rapid adaptation, making it suitable for dynamic scenarios like autonomous vehicles, medical imaging, and augmented reality. Key Meta-Learning Techniques are the following:

- Prototypical Networks: Facilitate few-shot classification by learning a metric space where classification is based on distances to class prototypes.
- Siamese Networks: Twin networks that map similar data points closer in feature space, are useful for tasks like forgery detection across languages and styles.
- Model-Agnostic Meta-Learning: Trains models to adapt quickly to new tasks with a few gradient steps, excelling in few-shot classification, regression, and reinforcement learning.
- Memory-Augmented Models: Use external memory mechanisms to rapidly encode and retrieve new information without extensive retraining.

The advantages of Meta-learning in CV are:

- Fast Adaptation: Quickly adapts to new tasks with limited data, essential for dynamic environments like drones or robotics.
- Data Efficiency: Leverages prior knowledge, reducing the need for extensive labelled data, critical in areas like medical imaging.
- Cross-Domain Learning: Enhances generalisation across different visual domains, aiding in knowledge transfer between tasks.
- Personalisation: Tailors models to individual preferences or unique environments, such as user-specific AR applications.

The applications of Meta-learning in CV include:

- Image Classification: Quickly identifies unseen categories with few-shot or zero-shot learning.
- Object Detection and Tracking: Enhances robustness to visual variations.
- Image Segmentation: Useful in medical imaging and autonomous driving.
- Facial Recognition: Adapts to new faces with minimal training data.
- Pose Estimation and Scene Understanding: Critical for robotics and AR applications.

Challenges in Meta-Learning:

- Scalability: Difficulty in handling large-scale datasets and high-dimensional CV tasks.
- Generalisation: Struggles to perform well on unseen tasks and domains.
- Computational Complexity: High resource requirements can limit applicability in constrained environments.
- Task Diversity: Developing diverse task sets for training is challenging but essential for real-world generalisation.
- optimisation Stability: Sensitive to hyperparameters and optimisation methods, requiring careful tuning.
- Interpretability: Deep meta-learning models lack transparency, complicating trust and usability.

7.1.4. Synchronisation in CPS

Synchronisation is critical for aligning the timing and interactions among subsystems, sensors, and actuators in CPS. In ML-based CV, the following strategies are used:

- Timestamping: Attaches precise time metadata to data packets to align heterogeneous data streams.
- Sensor Fusion: Integrates data from multiple sensors for accurate environmental representation, used in applications like autonomous vehicles and robotics.
- Real-Time Task Scheduling: Ensures low-latency, high-accuracy processing under strict resource and time constraints, crucial for autonomous vehicles, drones, and robotics.

These strategies collectively enhance the synchronisation and efficiency of CPS in ML-based CV applications.

7.1.5. Optimisation Approaches

In CPS, balancing computational efficiency and accuracy is key due to constraints like limited hardware resources, real-time processing needs, and high-accuracy demands. Various optimisation approaches help manage these challenges:

- Model Compression Techniques:
  - Pruning: Removes redundant neurons or connections to reduce model size without significantly impacting accuracy.
  - Quantisation: Lowers the precision of model weights, reducing memory usage and speeding up computation.
  - Knowledge Distillation: Transfers knowledge from a large, complex model (teacher) to a smaller, simpler one (student), maintaining similar accuracy while enhancing computational efficiency.
  - Low-rank Factorisation: Approximates weight matrices with lower-rank matrices, reducing parameters, speeding up training, and improving inference efficiency.
  - Transfer Learning: Reuses pre-trained models for related tasks, reducing the need for extensive data labelling and speeding up adaptation to new tasks.
- Lightweight Architectures:
  - MobileNet: Uses depthwise separable convolutions to create lightweight models, customisable for different trade-offs between latency and accuracy.
  - EfficientNets: Achieve high accuracy with fewer parameters and FLOPS, designed for efficient scaling while maintaining performance.
- Hardware Acceleration and optimisation:
  - Parallelism: Utilises GPUs and specialised hardware (e.g., TPUs) for faster processing, crucial for large-scale problems.
  - Inference Pipeline optimisation: Streamlines processing to meet real-time requirements, balancing trade-offs between accuracy, throughput, and latency.
- Data Augmentation:
  - Deeply Learned Augmentation: Uses deep learning models to generate data variations, enhancing robustness.
  - Feature-Level Augmentation: Alters specific data features (e.g., brightness, contrast) to improve generalisation.
  - Meta-learning: Learns to generate optimal augmentations based on data characteristics.
  - Data Synthesis: Using techniques like 3D modelling and GANs to generate synthetic data, increasing the diversity of the training set.
- Edge Computing:
  - Edge Intelligence: Combines AI and edge computing to address challenges such as reducing latency, optimising resources, and enhancing data privacy.
  - Distributed Learning: Collaborative model training across edge devices reduces bandwidth and enhances privacy, with local data storage requirements.

These approaches optimise computational resources, improve real-time performance, and maintain high accuracy in CPS applications like object detection, tracking, and decision-making.

7.1.6. Adaptation Mechanisms

In dynamic and unpredictable environments, CPS models need to be adaptable to changing conditions. Key adaptation mechanisms include:

- Data-driven Adaptation: Utilises real-world data to enable models to adjust to varying conditions, such as lighting, material properties, and geometric complexities. Techniques like data augmentation (geometric and photometric transformations) allow models to adapt to unseen scenarios and improve generalisation across diverse environments.

- Online Learning: Models are continuously updated with new data collected during deployment. This approach is critical for real-time adjustments to environmental changes, such as varying lighting or evolving cybersecurity threats. Examples include combining offline and online learning techniques, such as the Truncated Gradient Confidence-weighted model, for tasks like image classification.
- Transfer Learning: Involves fine-tuning pre-trained models on smaller task-specific datasets, enabling adaptation to new environments without training from scratch. This is especially useful in CPS where models trained in one context are adapted to others. For example, transfer learning is used to detect attacks in CPS using a ResNet, adapting the model to different operational conditions.
- Ensemble Methods: Combines predictions from multiple models to improve accuracy and robustness. By leveraging diverse architectures (e.g., MobileNetV2, ResNet50), ensemble methods enhance the model's ability to adapt to variations in data quality and features, as seen in medical image analysis tasks where data can differ in noise, resolution, and conditions.
- Adversarial Training: Enhances model resilience to small, intentional perturbations by incorporating adversarial examples into the training process. This helps models adapt to a broader range of input variations, making them more robust against adversarial attacks. Randomised smoothing further strengthens this by adding noise to the input, ensuring the model is resistant to adversarial modifications.
- Federated Learning: Allows distributed CPS devices to train models locally and share updates without centralising sensitive data. This improves performance while maintaining privacy. FL employs various aggregation methods (e.g. averaging, FedGKT) and privacy-preserving techniques (e.g. differential privacy, homomorphic encryption). Challenges include high communication overhead and non-IID data issues, but methods like Fed-MPS reduce resource constraints and communication costs.

These adaptation strategies ensure that CPS models can maintain high performance and resilience in dynamic, real-world environments by continuously learning, adapting, and optimising based on changing conditions.

7.2. Current Advancements and Future Directions

Figure 13 provides a comparative summary of challenges in synchronisation, optimisation, and adaptation, outlining current research, limitations, and potential future directions. It highlights areas such as real-time performance, architectural complexity, resource constraints, real-world deployment, and standardized benchmarks while suggesting federated learning, adaptive models, and emerging technologies (e.g., edge computing, hybrid model, neuromorphic computing, VPU, VFMs) to enhance scalability, efficiency, interoperability, and sustainability.

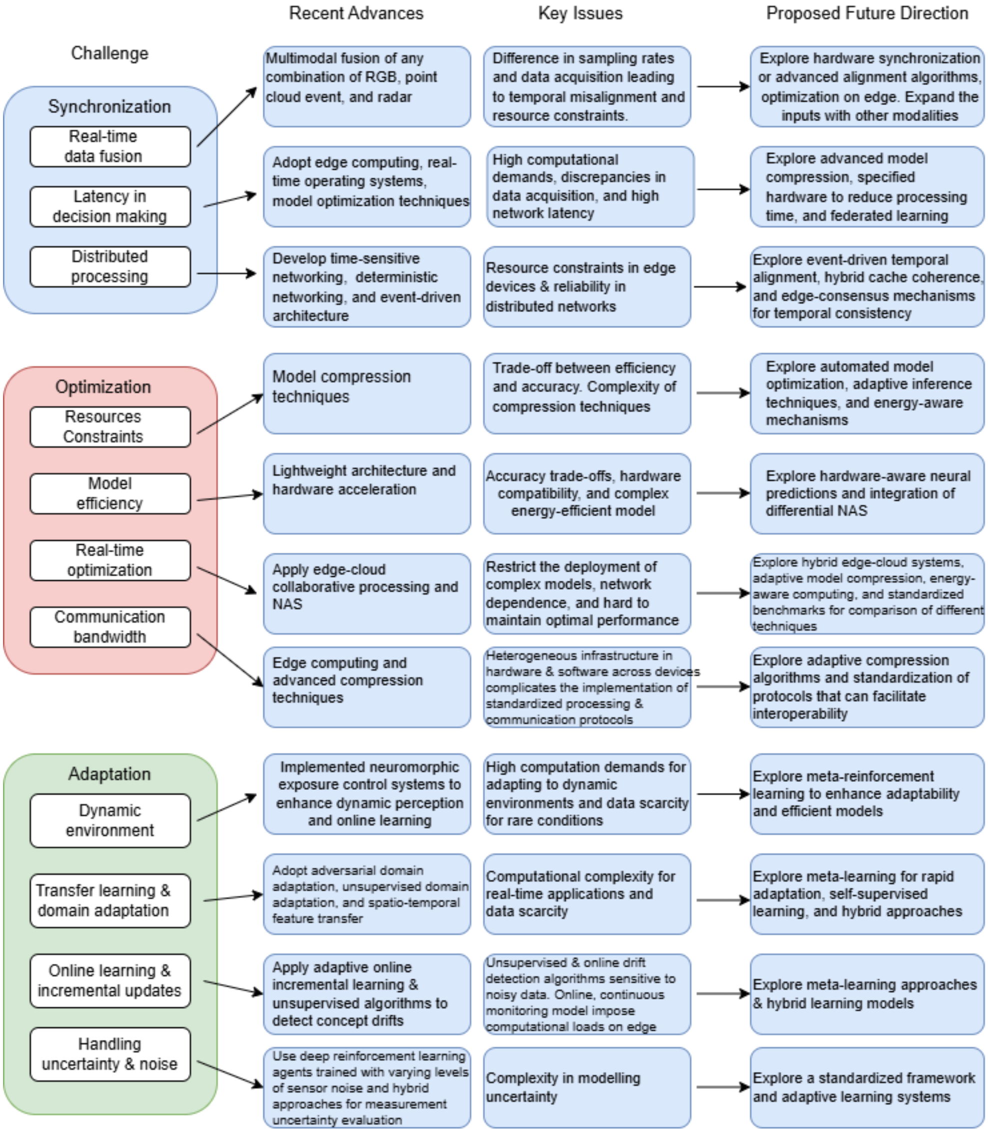
Recent Advances
Key Issues
Proposed Future Direction
Challenge
Synchronization
Real-time data fusion
Latency in decision making
Distributed processing
Multimodal fusion of any combination of RGB, point cloud event, and radar
Difference in sampling rates and data acquisition leading to temporal misalignment and resource constraints.
Explore hardware synchronization or advanced alignment algorithms, optimization on edge. Expand the inputs with other modalities
Adopt edge computing, real-time operating systems, model optimization techniques
High computational demands, discrepancies in data acquisition, and high network latency
Explore advanced model compression, specified hardware to reduce processing time, and federated learning
Develop time-sensitive networking, deterministic networking, and event-driven architecture
Resource constraints in edge devices & reliability in distributed networks
Explore event-driven temporal alignment, hybrid cache coherence, and edge-consensus mechanisms for temporal consistency
Optimization
Resources Constraints
Model efficiency
Real-time optimization
Communication bandwidth
Model compression techniques
Trade-off between efficiency and accuracy. Complexity of compression techniques
Explore automated model optimization, adaptive inference techniques, and energy-aware mechanisms
Lightweight architecture and hardware acceleration
Accuracy trade-offs, hardware compatibility, and complex energy-efficient model
Explore hardware-aware neural predictions and integration of differential NAS
Apply edge-cloud collaborative processing and NAS
Restrict the deployment of complex models, network dependence, and hard to maintain optimal performance
Explore hybrid edge-cloud systems, adaptive model compression, energy-aware computing, and standardized benchmarks for comparison of different techniques
Edge computing and advanced compression techniques
Heterogeneous infrastructure in hardware & software across devices complicates the implementation of standardized processing & communication protocols
Explore adaptive compression algorithms and standardization of protocols that can facilitate interoperability
Adaptation
Dynamic environment
Transfer learning & domain adaptation
Online learning & incremental updates
Handling uncertainty & noise
Implemented neuromorphic exposure control systems to enhance dynamic perception and online learning
High computation demands for adapting to dynamic environments and data scarcity for rare conditions
Explore meta-reinforcement learning to enhance adaptability and efficient models
Adopt adversarial domain adaptation, unsupervised domain adaptation, and spatio-temporal feature transfer
Computational complexity for real-time applications and data scarcity
Explore meta-learning for rapid adaptation, self-supervised learning, and hybrid approaches
Apply adaptive online incremental learning & unsupervised algorithms to detect concept drifts
Unsupervised & online drift detection algorithms sensitive to noisy data. Online, continuous monitoring model impose computational loads on edge
Explore meta-learning approaches & hybrid learning models
Use deep reinforcement learning agents trained with varying levels of sensor noise and hybrid approaches for measurement uncertainty evaluation
Complexity in modelling uncertainty
Explore a standardized framework and adaptive learning systems

Figure 13. Roadmap for Future Research.

The integration of ML for computer vision into CPS is revolutionising industries by enabling smarter, more adaptable, and efficient operations. Techniques such as edge AI, self-adaptive RL, and hybrid models exemplify the advancements in CPS, addressing challenges in real-time processing, scalability, and decision-making. These approaches enable applications like healthcare, autonomous vehicles, and smart cities to function more effectively. However, significant challenges remain in scalability, adaptability, and deployment, underscoring the importance of continued research and interdisciplinary collaboration.

Edge AI enhances CPS by performing ML tasks directly at the network edge, allowing for real-time decision-making with reduced latency and improved privacy. By processing data locally, it minimises reliance on centralised servers, making it ideal for applications requiring rapid responses, such as industrial automation and autonomous systems. Self-adaptive RL contributes by equipping CPS with the ability to adapt to dynamic environments through learning from interactions rather than static datasets. This capability is particularly valuable in robotics and IoT healthcare, where real-time decision-making and resource optimisation are critical. Hybrid models further advance CPS by combining deep learning's feature extraction capabilities with the interpretability of traditional algorithms, creating efficient and scalable solutions for complex tasks like language translation and anomaly detection.

Despite these advancements, significant challenges persist. Scalability issues arise as many synchronisation methods assume abundant resources, which are not always available in real-world CPS. Lightweight algorithms that balance accuracy, energy efficiency, and computational costs are urgently needed. Additionally, CPS systems must adapt to dynamic conditions such as environmental variability and real-time constraints, challenges that current methods are often ill-equipped to handle. Technologies like neuromorphic computing and event-based vision require new synchronisation strategies to unlock their potential effectively. Continued research is essential to address these gaps, developing innovative algorithms and techniques that ensure CPS systems can meet the demands of diverse and evolving applications.

Interdisciplinary collaboration plays a crucial role in advancing CPS technologies. By bringing together expertise from fields like computer science, engineering, and domain-specific disciplines, researchers can develop solutions that are both technically sound and practically viable. For example, collaboration is essential in integrating emerging technologies such as FL and edge computing with CPS. Edge computing enhances efficiency through strategies like model optimisation, workload partitioning, and real-time monitoring, while FL supports decentralised model training, preserving privacy and enabling robust collaboration across devices. Together, these approaches ensure that CPS systems are not only innovative but also resilient, secure, and scalable.

Ethical and regulatory considerations further highlight the importance of interdisciplinary efforts. Establishing standardised benchmarks fosters consistent evaluation and collaboration. Sustainability and inclusivity must also be prioritized to address environmental concerns and broaden access to these transformative technologies. By combining technical innovation with interdisciplinary collaboration and ethical practices, CPS can continue to evolve, delivering scalable, secure, and responsible solutions that meet the needs of modern industries and society.

Author Contributions: Kai Hung Tang: Methodology, Writing – original draft, Writing–review & editing. **Mohamed Chahine Ghanem:** Methodology, Supervision, Writing–review & editing. **Pawel Gasiorowski:** Methodology, Supervision, Writing–review & editing. **Vassil Vassilev:** Methodology & Editing. **Karim Ouazzane:** Methodology & Editing.

Funding: The APC and Open Access fees for this work are funded by the University of Liverpool.

Institutional Review Board Statement: This research was deemed not to require ethical approval.

Data Availability Statement: No data was used for the research described in the article.

Conflicts of Interest: The authors declare that they have no known competing interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*.
2. Phung, V.H.; Rhee, E.J. A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences* **2019**, *9*.
3. Mirzaei, S.; Kang, J.L.; Chu, K.Y. A comparative study on long short-term memory and gated recurrent unit neural networks in fault diagnosis for chemical processes using visualization. *Journal of the Taiwan Institute of Chemical Engineers* **2022**, *130*.
4. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv* **2020**, *abs/2010.11929*.
5. Esterle, L.; Grosu, R. Cyber-physical systems: challenge of the 21st century. *Elektrotech. Inftech* **2016**, pp. 299–303.
6. Jahromi Babak Shahian, T.T.; Sabri, C. Real-Time Hybrid Multi-Sensor Fusion Framework for Perception in Autonomous Vehicles. *Sensors* **2018**, *20*, 4357.

7. Bennett, T.R.; Gans, N.; Jafari, R. A data-driven synchronization technique for cyber-physical systems. In Proceedings of the Proceedings of the Second International Workshop on the Swarm at the Edge of the Cloud, New York, NY, USA, 2015; SWEC '15, p. 49–54.
8. Wu, H.; Zeng, L.; Chen, M.; Wang, T.; He, C.; Xiao, H.; Luo, S. Weak surface defect detection for production-line plastic bottles with multi-view imaging system and LFF YOLO. *Optics and Lasers in Engineering* **2024**, *181*, 108369.
9. Robyns, S.; Heerwegh, W.; Weckx, S. A Digital Twin of an Off Highway Vehicle based on a Low Cost Camera. *Procedia Computer Science* **2024**, *232*, 2366–2375. 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023).
10. Reddy, G.J.; Sharma, D.S.G. Edge AI in Autonomous Vehicles: Navigating the Road to Safe and Efficient Mobility. *International Journal of Scientific Research in Engineering and Management* **2024**, *08*, 1—13.
11. Maier, G.; Pfaff, F.; Wagner, M.; Pieper, C.; Gruna, R.; Noack, B.; Kruggel-Emden, H.; Längle, T.; Hanebeck, U.D.; Wirtz, S.; et al. Real-time multitarget tracking for sensor-based sorting. *Journal of Real-Time Image Processing* **2019**, *16*, 2261–2272.
12. Wang, S.; Zhang, J.; Wang, P.; Law, J.; Calinescu, R.; Mihaylova, L. A deep learning-enhanced Digital Twin framework for improving safety and reliability in human–robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing* **2024**, *85*, 102608.
13. Prasad, K.P.S.P. Compressed Mobilenet V3: An Efficient CNN for Resources Constrained Platforms. *Purdue University Graduate School. Thesis.* **2021**.
14. Wang, S.; Zheng, H.; Wen, X.; Shang, F. Distributed High-Performance Computing Methods for Accelerating Deep Learning Training. *Journal of Knowledge Learning and Science Technology* **2024**, *3*.
15. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv* **2016**.
16. Bird, J.J.; Kobylarz, J.; Faria, D.R.; Ekárt, A.; Ribeiro, E.P. Cross-Domain MLP and CNN Transfer Learning for Biological Signal Processing: EEG and EMG. *IEEE Access* **2020**, *8*, 54789–54801.
17. Hanhirova, J.; Kämäräinen", T.; Seppälä, S.; Siekkinen, M.; Hirvisalo, V.; Ylä-Jääski, A. Latency and throughput characterization of convolutional neural networks for mobile computer vision. *MMSys '18: Proceedings of the 9th ACM Multimedia Systems Conference* **2018**, pp. 204–215.
18. Shen, Y.; Li, Y.; Liu, Y.; Wang, Y.; Chen, L.; Wang, F.Y. Conditional visibility aware view synthesis via parallel light fields. *Neurocomputing* **2024**, *588*, 127644.
19. Kaur, P.; Khehra, B.S.; Mavi, E.B.S. Data Augmentation for Object Detection: A Review. In Proceedings of the 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021, pp. 537–543.
20. Wang, H.; Zhang, H.; Zhu, L.; Wang, Y.; Deng, J. ResADM: A Transfer-Learning-Based Attack Detection Method for Cyber–Physical Systems. *Applied Sciences* **2023**, *13*, 13019.
21. Awan, A.A. What is Online Machine Learning? *datacamp.com/blog/what-is-online-machine-learning* **2023**.
22. Tahir, A.; Saadia, A.; Khan, K.; Gul, A.; Qahmash, A.; Akram, R. Enhancing diagnosis: ensemble deep-learning model for fracture detection using X-ray images. *Clinical Radiology* **2024**, *79*, 1394–1402.
23. Liang, F.; Wu, B.; Wang, J.; Yu, L.; Li, K.; Zhao, Y.; Misra, I.; Huang, J.B.; Zhang, P.; Vajda, P.; et al. FlowVid: Taming Imperfect Optical Flows for Consistent Video-to-Video Synthesis. *arXiv* **2023**, [2312.17681].
24. Bengar, J.Z.; Gonzalez-Garcia, A.; Villalonga, G.; Raducanu, B.; Aghdam, H.H.; Mozerov, M. Temporal Coherence for Active Learning in Videos. *arXiv* **2019**.
25. Dantas, P.V.; da Silva Jr, W.S.; Cordeiro, L.C.; Carvalho, C.b. A comprehensive review of model compression techniques in machine learning. *Application Intelligence* **2024**, *54*, 11804–11844.
26. Cai, E.; Juan, D.C.; Stamoulis, D.; Marculescu, D. NeuralPower: Predict and Deploy Energy-Efficient Convolutional Neural Networks. *arXiv:1710.05420* **2017**.
27. He, K.; Gkioxari, G.; Dollär, P.; Girshick, R. Mask R-CNN. *arxiv.org/1703.06870v3* **2018**.
28. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *Computer Vision - ECCV 2016. Lecture Notes in Computer Science ()* **2016**, *9905*, 21–37.
29. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arxiv:2005.12872v3* **2020**.
30. Hu, J.; Yan, C.; Liu, X.; Li, Z.; Ren, C.; Zhang, J.; Peng, D.; Yang, Y. An integrated classification model for incremental learning. *Multimedia Tools and Applications* **2021**, *80*, 17275–17290.
31. Shoukat, M.U.; Yan, L.; Yan, Y.; Zhang, F.; Zhai, Y.; Han, P.; Nawaz, S.A.; Raza, M.A.; Akbar, M.W.; Hussain, A. Autonomous driving test system under hybrid reality: The role of digital twin technology. *Internet of Things* **2024**, *27*, 101301.

32. Pan, Y.; Luo, K.; Liu, Y.; Xu, C.; Liu, Y.; Zhang, L. Mobile edge assisted multi-view light field video system: Prototype design and empirical evaluation. *Future Generation Computer Systems* **2024**, *153*, 154–168.
33. Sakina, M.; Muhammad, I.; Abdullahi, S.S. A multi-factor approach for height estimation of an individual using 2D image. *Procedia Computer Science* **2024**, *231*, 765–770.
34. Kaushik, H.; Kumar, T.; Bhalla, K. iSecureHome: A deep fusion framework for surveillance of smart homes using real-time emotion recognition. *Applied Soft Computing* **2022**, *122*, 108788.
35. Murel, J.; Kavlakoglu, E. What is object detection? *ibm.com/topics/object-detection* **2024**.
36. GreeksforGreeks. What is Object Detection in Computer Vision? *geeksforgeeks.org/what-is-object-detection-in-computer-vision/* **2024**.
37. Himeur, Y.; Varlamis, I.; Kheddar, H.; Amira, A.; Atalla, S.; Singh, Y.; Bensaali, F.; Mansoor, W. Federated Learning for Computer Vision. *arXiv:2308.13558v1* **2023**.
38. Sprague, M.R.; Jalalirad, A.; Scavuzzo, M.; Capota, C.; Neun, M.; Do, L.; Kopp, M. Asynchronous Federated Learning for Geospatial Applications. *Communications in Computer and Information Science* **2019**, *967*.
39. He, C.; Shah, A.D.; Tang, Z.; Sivashunmugam, D.F.N.; Bhogaraju, K.; Shimpi, M.; Shen, L.; Chu, X.; Soltanolkotabi, M.; Avestimehr, S. FedCV: A Federated Learning Framework for Diverse Computer Vision Tasks. *Computer Vision and Pattern Recognition* **2021**.
40. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. *arXiv* **2017**.
41. Dey, S.; Dutta, A.; Toledo, J.I.; Ghosh, S.K.; Lladós, J.; Pal, U. SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. *ArXiv* **2017**, *abs/1707.02131*.
42. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv* **2017**.
43. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-Learning with Memory-Augmented Neural Networks. In Proceedings of the Proceedings of The 33rd International Conference on Machine Learning; Balcan, M.F.; Weinberger, K.Q., Eds., New York, New York, USA, 20–22 Jun 2016; Vol. 48, *Proceedings of Machine Learning Research*, pp. 1842–1850.
44. Golovin, D.; Solnik, B.; Moitra, S.; Kochanski, G.; Karro, J.E.; Sculley, D., Eds. *Google Vizier: A Service for Black-Box Optimization*, 2017.
45. Yang, H.s.; Kupferschmidt, B. Time Stamp Synchronization in Video System. *International Telemetering Conference Proceedings* **2010**.
46. Lin, Z.; Li, P.; Xiao, Y.; Xiao, L.; Luo, F. Learning Based Efficient Federated Learning for Object Detection in MEC Against Jamming. In Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC), 2021, pp. 115–120. https://doi.org/10.1109/ICCC52777.2021.9580318.
47. Hu, Y.; Liu, S.; Abdelzaher, T.; Wigness, M.; David, P. Real-time task scheduling with image resizing for criticality-based machine perception. *Real-Time Systems* **2022**, *58*, 430–455.
48. Ben-Num, T.; Hoefler, T. Demystifying Parallel and Distributed Deep Learning: An In-depth Concurrency Analysis. *ACM Computing Surveys (CSUR)* **2019**, *52*, 1–43.
49. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**.
50. Cai, G.; Li, J.; Liu, X.; Chen, Z.; Zhang, H. Learning and Compressing: Low-Rank Matrix Factorization for Deep Neural Network Compression. *Applied Sciences* **2023**, *13*.
51. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**.
52. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**.
53. Yan, Z.; Bai, Z.; Wong, W.F. Reconsidering the energy efficiency of spiking neural networks, 2024, [arXiv:cs.NE/2409.08290].
54. Mumuni, A.; Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. *Array* **2022**, *16*, 100258.
55. Deng, S.; Zhao, H.; Fang, W.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *IEEE Internet of Things Journal* **2020**, *7*, 7457–7469.
56. Tron, R.; Vidal, R. Distributed computer vision algorithms through distributed averaging. In Proceedings of the CVPR 2011, 2011, pp. 57–63.
57. Lu, J.; Zhao, P.; Hoi, S.C.H. Online Passive-Aggressive Active Learning. *Machine Learning* **2016**, *103*, 141–183.
58. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**.
59. Cohen, J.M.C.; Rosenfeld, E.; Kolter, J.Z. Certified Adversarial Robustness via Randomized Smoothing. *arXiv* **2019**.

60. Jiang, S.; Wang, X.; Que, Y.; Lin, H. Fed-MPS: Federated learning with local differential privacy using model parameter selection for resource-constrained CPS. *Journal of Systems Architecture* **2024**, *150*.
61. Nazare Jr., A.C.; Schwartz, W.R. A scalable and flexible framework for smart video surveillance. *Computer Vision and Image Understanding* **2016**, *144*, 258–275. Individual and Group Activities in Video Event Analysis, https://doi.org/https://doi.org/10.1016/j.cviu.2015.10.014.
62. Gallego, G.; Delbruck, T.; Orchard, G.; Bartolozzi, C.; Taba, B.; Censi, A.; Leutenegger, S.; Davison, A.J.; Conradt, J.; Daniilidis, K.; et al. Event-Based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2022**, *44*, 154—180. https://doi.org/10.1109/tpami.2020.3008413.
63. Ren, H.; Zhou, Y.; Huang, Y.; Fu, H.; Lin, X.; Song, J.; Cheng, B. SpikePoint: An Efficient Point-based Spiking Neural Network for Event Cameras Action Recognition, 2024, [arXiv:cs.CV/2310.07189].
64. Ji, M.; Wang, Z.; Yan, R.; Liu, Q.; Xu, S.; Tang, H. SCTN: Event-based object tracking with energy-efficient deep convolutional spiking neural networks. *Frontiers in Neuroscience* **2023**, *Volume 17 - 2023*. https://doi.org/10.3389/fnins.2023.1123698.
65. Cordone, L.; Miramond, B.; Ferrante, S. Learning from Event Cameras with Sparse Spiking Convolutional Neural Networks, 2021, [arXiv:cs.CV/2104.12579].
66. Brödermann, T.; Bruggemann, D.; Sakaridis, C.; Gool, L.V. EFCL Dataset - Technical Report. Technical report, ETH Zurich, 2022.
67. Xing, W.; Lin, S.; Yang, L.; Pan, J. Target-free Extrinsic Calibration of Event-LiDAR Dyad using Edge Correspondences, 2023, [arXiv:cs.RO/2305.04017].
68. Zhou, X.; Dai, Y.; Qin, H.; Qiu, S.; Liu, X.; Dai, Y.; Li, J.; Yang, T. Subframe-Level Synchronization in Multi-Camera System Using Time-Calibrated Video. *Sensors* **2024**, *24*. https://doi.org/https://doi.org/10.3390/s24216975.
69. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3, 2019, [arXiv:cs.CV/1905.02244].
70. Senadeera, D.C.; Yang, X.; Kollias, D.; Slabaugh, G. CUE-Net: Violence Detection Video Analytics with Spatial Cropping, Enhanced UniformerV2 and Modified Efficient Additive Attention, 2024, [arXiv:cs.CV/2404.18952].
71. cob parro, a.c.; losada gutiérrez, c.; marrón romera, m.; gardel vicente, a.; bravo muñoz, i. smart video surveillance system based on edge computing. *sensors* **2021**, *21*. https://doi.org/10.3390/s21092958.
72. Wan, Z.; Mao, Y.; Zhang, J.; Dai, Y. RPEFlow: Multimodal Fusion of RGB-PointCloud-Event for Joint Optical Flow and Scene Flow Estimation, 2023, [arXiv:cs.CV/2309.15082].
73. Pei, K.; Zhu, L.; Cao, Y.; Yang, J.; Vondrick, C.; Jana, S. Towards Practical Verification of Machine Learning: The Case of Computer Vision Systems, 2022, [arXiv:cs.CR/1712.01785].
74. Santos, V.f.; Albuquerque, C.; Passos, D.; Quincozes, S.E.; Mossé, D. Assessing Machine Learning Techniques for Intrusion Detection in Cyber-Physical Systems. *Energies* **2023**, *16*. https://doi.org/10.3390/en16166058.
75. Windmann, A.; Steude, H.; Niggemann, O. Robustness and Generalization Performance of Deep Learning Models on Cyber-Physical Systems: A Comparative Study, 2023, [arXiv:cs.LG/2306.07737].
76. Calba, C.; Goutard, F.L.; Hoinville, L.; Hendrikx, P.; Lindberg, A.; Saegerman, C.; Peyre, M. Surveillance systems evaluation: a systematic review of the existing approaches. *BMC Public Health* **2015**, *15*. https://doi.org/10.1186/s12889-015-1791-5.
77. Kim, Y.; Jeong, J. A Simulation-Based Approach to Evaluate the Performance of Automated Surveillance Camera Systems for Smart Cities. *Applied Science* **2023**, *13*. https://doi.org/10.3390/app131910682.
78. Pagire, V.; Chavali, M.; Kale, A. A comprehensive review of object detection with traditional and deep learning methods. *Signal Processing* **2025**, *237*, 110075. https://doi.org/https://doi.org/10.1016/j.sigpro.2025.110075.
79. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.l.; Da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*. https://doi.org/10.3390/electronics10030279.
80. Luo, C.; He, X.; Zhan, J.; Wang, L.; Gao, W.; Dai, J. Comparison and Benchmarking of AI Models and Frameworks on Mobile Devices, 2020, [arXiv:cs.LG/2005.05085].
81. Park, H.D.; Min, O.G.; Lee, Y.J. Scalable Architecture for an Automated Surveillance System Using Edge Computing. *J Supercomput* **2017**.
82. Boizard, N.; Haddad, K.E.; Ravet, T.; Cresson, F.; Dutoit, T. Deep learning-based stereo camera multi-video synchronization, 2023, [arXiv:cs.CV/2303.12916].

83. Michaelis, C.; Mitzkus, B.; Geirhos, R.; Rusak, E.; Bringmann, O.; Ecker, A.S.; Bethge, M.; Brendel, W. Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming, 2020, [arXiv:cs.CV/1907.07484].
84. Zheng, Z.; Cheng, Y.; Xin, Z.; Yu, Z.; Zheng, B. Robust Perception Under Adverse Conditions for Autonomous Driving Based on Data Augmentation. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 13916–13929.
85. Maróti, M.; Kusy, B.; Simon, G.; Lédeczi, A. The flooding time synchronization protocol. *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems* **2004**, pp. 33–49.
86. Abdul-Rashid, R.; Al-Shaikhi, A.; Masoud, A. Accurate, Energy-Efficient, Decentralized, Single-Hop, Asynchronous Time Synchronization Protocols for Wireless Sensor Networks, 2018, [arXiv:eess.SP/1811.01152].
87. Kreković, D.; Krivić, P.; Žarko, I.P.; Kušek, M.; Le-Phuoc, D. Reducing Communication Overhead in the IoT-Edge-Cloud Continuum: A Survey on Protocols and Data Reduction Strategies, 2024, [arXiv:cs.NI/2404.19492].
88. Kyriakakis, E.; Tange, K.; Reusch, N.; Zaballa, E.O.; Fafoutis, X.; Schoeberl, M.; Dragoni, N. Fault-tolerant Clock Synchronization using Precise Time Protocol Multi-Domain Aggregation. In Proceedings of the 2021 IEEE 24th International Symposium on Real-Time Distributed Computing (ISORC), 2021, pp. 114–122.
89. Gore, R.N.; Lisova, E.; Åkerberg, J.; Björkman, M. CoSiNeT: A Lightweight Clock Synchronization Algorithm for Industrial IoT. In Proceedings of the 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), 2021, pp. 92–97.
90. Yao, Z.; Jia, L.; Wang, Y.; Wang, Z.; Zhou, Z.; Liao, B.; Mumtaz, S.; Wang, X. Time Synchronization-Aware Edge-End Collaborative Network Routing Management for FL-Assisted Distributed Energy Scheduling. In Proceedings of the ICC 2023 - IEEE International Conference on Communications, 2023, pp. 1780–1785.
91. Jiang, C.; Fan, T.; Gao, H.; Shi, W.; Liu, L.; Cérin, C.; Wan, J. Energy aware edge computing: A survey. *Computer Communications* **2020**, *151*, 556–580.
92. Vasile, C.E.; Ulmămei, A.A.; Bîră, C. Image Processing Hardware Acceleration—A Review of Operations Involved and Current Hardware Approaches. *Journal of Imaging* **2024**, *10*.
93. Piccolboni, L.; Mantovani, P.; Guglielmo, G.D.; Carloni, L.P. COSMOS: Coordination of High-Level Synthesis and Memory Optimization for Hardware Accelerators. *ACM Trans. Embed. Comput. Syst.* **2017**, *16*.
94. Rehman, M.; Petrillo, A.; Forcina, A.; Felice, F.D. Metaverse Simulator for Emotional Understanding. *Procedia Computer Science* **2024**, *232*, 3216–3228.
95. Adnan Arefeen, M.; Tabassum Nimi, S.; Yusuf Sarwar Uddin, M. FrameHopper: Selective Processing of Video Frames in Detection-driven Real-Time Video Analytics. In Proceedings of the 2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2022, pp. 125–132.
96. Zhi, Y.; Tong, Z.; Wang, L.; Wu, G. MGSampler: An Explainable Sampling Strategy for Video Action Recognition, 2021, [arXiv:cs.CV/2104.09952].
97. Li, J.; Xu, R.; Liu, X.; Ma, J.; Li, B.; Zou, Q.; Ma, J.; Yu, H. Domain Adaptation based Object Detection for Autonomous Driving in Foggy and Rainy Weather. *IEEE Transactions on Intelligent Vehicles* **2024**, pp. 1–12.
98. Wang, Y.; Lu, Y.; Qiu, Y. Gated image-adaptive network for driving-scene object detection under nighttime conditions. *Multimedia Syst.* **2024**, *31*.
99. Madasu, A.; Vijjini, A.R. Sequential Domain Adaptation through Elastic Weight Consolidation for Sentiment Analysis. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 4879–4886.
100. Aslam, S.; Rasool, A.; Wu, H.; Li, X. CEL: A Continual Learning Model for Disease Outbreak Prediction by Leveraging Domain Adaptation via Elastic Weight Consolidation, 2024, [arXiv:cs.LG/2401.08940].
101. Zhou, Z.; Yeung, G.; Schapiro, A.C. Self-recovery of memory via generative replay, 2023, [arXiv:cs.NE/2301.06030].
102. Lv, J.; Li, X.; Sun, Y.; Zheng, Y.; Bao, J. A bio-inspired LIDA cognitive-based Digital Twin architecture for unmanned maintenance of machine tools. *Robotics and Computer–Integrated Manufacturing* **2023**, p. 102489.
103. Tong, H.; Chen, M.; Zhao, J.; Hu, Y.; Yang, Z.; Liu, Y.; Yin, C. Continual Reinforcement Learning for Digital Twin Synchronization Optimization, 2025, [arXiv:cs.NI/2501.08045].
104. García-Valls, M.; Chirivella-Ciruelos, A.M. CoTwin: Collaborative improvement of digital twins enabled by blockchain. *Future Generation Computer Systems* **2024**, *157*, 408–421.
105. Scarrica, V.M.; Panariello, C.; Ferone, A.; Staiano, A. A Hybrid Approach To Real-Time Multi-Object Tracking, 2023, [arXiv:cs.CV/2308.01248].

106. Ussa, A.; Rajen, C.S.; Pulluri, T.; Singla, D.; Acharya, J.; Chuanrong, G.F.; Basu, A.; Ramesh, B. A Hybrid Neuromorphic Object Tracking and Classification Framework for Real-Time Systems. *IEEE Transactions on Neural Networks and Learning Systems* **2024**, *35*, 10726–10735.
107. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need, 2023, [arXiv:cs.CL/1706.03762].
108. Al-Qawlaq, A.; M, A.K.; John, D. KWT-Tiny: RISC-V Accelerated, Embedded Keyword Spotting Transformer, 2024, [arXiv:cs.AR/2407.16026].
109. Wang, W.; Sun, W.; Liu, Y. Improving Transformer Inference Through Optimized Non-Linear Operations With Quantization-Approximation-Based Strategy. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2024**, pp. 1–1.
110. Piardi, L.; Leitão, P.; Queiroz, J.; Pontes, J. Role of digital technologies to enhance the human integration in industrial cyber–physical systems. *Annual Reviews in Control* **2024**, *57*, 100934.
111. Wang, B.; Zheng, P.; Yin, Y.; Shih, A.; Wang, L. Toward human-centric smart manufacturing: A human-cyber-physical systems (HCPS) perspective. *Journal of Manufacturing Systems* **2022**, *63*, 471–490.
112. Xia, L.; Lu, J.; Lu, Y.; Hao, Z.; Fan, Y.; Zhang, Z. Augmented reality and indoor positioning based mobile production monitoring system to support workers with human-in-the-loop. *Robotics and Computer-Integrated Manufacturing* **2024**, *86*, 102664.
113. Xiao, W.; Li, A.; Cassandras, C.G.; Belta, C. Toward model-free safety-critical control with humans in the loop. *Annual Reviews in Control* **2024**, *57*, 100944.
114. Saeed, M.M.; Alsharidah, M. Security, privacy, and robustness for trustworthy AI systems: A review. *Computers and Electrical Engineering* **2024**, *119*, 109643. https://doi.org/https://doi.org/10.1016/j.compeleceng.2024.109643.
115. Tahir, G.A. Ethical Challenges in Computer Vision: Ensuring Privacy and Mitigating Bias in Publicly Available Datasets, 2024, [arXiv:cs.CV/2409.10533].
116. Khargonekar, P.P.; Sampath, M. A Framework for Ethics in Cyber-Physical-Human Systems. *IFAC-PapersOnLine* **2020**, *53*, 17008–17015. 21st IFAC World Congress, https://doi.org/https://doi.org/10.1016/j.ifacol.2020.12.1251.
117. Zahidi, S.; Ratcheva, V.; Hingel, G.; Brown, S. The Future of Jobs Report 2020. *World Economic Forum* **2020**.
118. Broo, D.G. Transdisciplinarity and three mindsets for sustainability in the age of cyber-physical systems. *Journal of Industrial Information Integration* **2022**, *27*, 100290.
119. Gerhold, M.; Kouzel, A.; Mangal, H.; Mehmed, S.; Zaytsev, V. Modelling of Cyber-Physical Systems through Domain-Specific Languages: Decision, Analysis, Design. *MODELS Companion '24: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems* **2024**, pp. 1170–1179.
120. Ghanem, M.; Dawoud, F.; Gamal, H.; Soliman, E.; El-Batt, T.; Sharara, H. FLoBC: A decentralized blockchain-based federated learning framework. In Proceedings of the 2022 Fourth International Conference on Blockchain Computing and Applications (BCCA). IEEE, 2022, pp. 85–92.
121. Ghanem, M.; Mouloudi, A.; Mourchid, M. Towards a scientific research based on semantic web. *Procedia Computer Science* **2015**, *73*, 328–335.
122. Zhao, Y.; Zhu, Z.; Chen, B.; Qiu, S.; Huang, J.; Lu, X.; Yang, W.; Ai, C.; Huang, K.; He, C.; et al. Toward parallel intelligence: An interdisciplinary solution for complex systems. *The Innovation* **2023**, *4*, 100521.
123. Tomelleri, F.; Sbaragli, A.; Piacariello, F.; Pilati, F. Safe Assembly in Industry 5.0: Digital Architecture for the Ergonomic Assembly Worksheet. *Procedia CIRP* **2024**, *127*, 68–73.
124. Ghanem, M.C.; Ratnayake, D.N. Enhancing WPA2-PSK four-way handshaking after re-authentication to deal with de-authentication followed by brute-force attack a novel re-authentication protocol. In Proceedings of the 2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA). IEEE, 2016, pp. 1–7.
125. Loumachi, F.Y.; Ghanem, M.C.; Ferrag, M.A. Advancing Cyber Incident Timeline Analysis Through Retrieval-Augmented Generation and Large Language Models. *Computers* **2025**, *14*, 67.
126. Yedilkhan, D.; Kyzyrkanov, A.E.; Kutpanova, Z.A.; Aljawarneh, S.; Atanov, S.K. Intelligent obstacle avoidance algorithm for safe urban monitoring with autonomous mobile drones. *Journal of Electronic Science and Technology* **2024**, *22*, 100277.
127. Towards an efficient automation of network penetration testing using model-based reinforcement learning. PhD thesis, City, University of London, 2022.

128. Ghanem, M.C.; Salloum, S. Integrating AI-Driven Deep Learning for Energy-Efficient Smart Buildings in Internet of Thing-based Industry 4.0. *Babylonian Journal of Internet of Things* **2025**, *2025*, 121–130.
129. Meuser, T.; Lovén, L.; Bhuyan, M.; Patil, S.G.; Dustdar, S.; Aral, A.; Bayhan, S.; Becker, C.; Lara, E.d.; Ding, A.Y.; et al. Revisiting Edge AI: Opportunities and Challenges. *IEEE Internet Computing* **2024**, *28*, 49—59.
130. Farzaan, M.A.; Ghanem, M.C.; El-Hajjar, A.; Ratnayake, D.N. AI-powered system for an efficient and effective cyber incidents detection and response in cloud environments. *IEEE Transactions on Machine Learning in Communications and Networking* **2025**.
131. Basnet, A.S.; Ghanem, M.C.; Dunsin, D.; Kheddar, H.; Sowinski-Mydlarz, W. Advanced persistent threats (apt) attribution using deep reinforcement learning. *Digital Threats: Research and Practice*.
132. Rocha, A.; Monteiro, M.; Mattos, C.; Dias, M.; Soares, J.; Magalhães, R.; Macedo, J. Edge AI for Internet of Medical Things: A literature review. *Computers and Electrical Engineering* **2024**, *116*, 109202. https://doi.org/https://doi.org/10.1016/j.compeleceng.2024.109202.
133. Alshinwan, M.; Memon, A.G.; Ghanem, M.C.; Almaayah, M. Unsupervised text feature selection approach based on improved Prairie dog algorithm for the text clustering. *Jordanian Journal of Informatics and Computing* **2025**, *2025*, 27–36.
134. Feit, F.; Metzger, A.; Pohl, K. Explaining Online Reinforcement Learning Decisions of Self-Adaptive Systems. *arXiv* **2022**.
135. Zhou, X.; Yang, J.; Li, Y.; Li, S.; Su, Z. Deep reinforcement learning-based resource scheduling for energy optimization and load balancing in SDN-driven edge computing. *Computer Communications* **2024**, *226-227*, 107925.
136. Singh, R.; Bengani, V. Hybrid Learning Systems: Integrating Traditional Machine Learning With Deep Learning Techniques. *ResearchGate* **2024**.
137. Kadhim, Y.A.; Guzel, M.S.; Mishra, A. A Novel Hybrid Machine Learning-Based System Using Deep Learning Techniques and Meta-Heuristic Algorithms for Various Medical Datatypes Classification. *Diagnostics* **2024**, *14*.
138. Jia, J.; Liang, W.; Liang, Y. A Review of Hybrid and Ensemble in Deep Learning for Natural Language Processing. *arXiv* **2023**.
139. et. al., R.B. On the Opportunities and Risks of Foundation Models, 2022, [arXiv:cs.LG/2108.07258].
140. Awais, M.; Naseer, M.; Khan, S.; Anwer, R.M.; Cholakkal, H.; Shah, M.; Yang, M.H.; Khan, F.S. Foundational Models Defining a New Era in Vision: A Survey and Outlook, 2023, [arXiv:cs.CV/2307.13721].
141. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision, 2021, [arXiv:cs.CV/2103.00020].
142. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything, 2023, [arXiv:cs.CV/2304.02643].
143. Zhao, X.; Ding, W.; An, Y.; Du, Y.; Yu, T.; Li, M.; Tang, M.; Wang, J. Fast Segment Anything, 2023, [arXiv:cs.CV/2306.12156].
144. Zhang, C.; Cho, J.; Puspitasari, F.D.; Sheng Zheng, Chenghao Li, Y.Q.T.K.X.S.C.Z.C.Q.F.R.L.H.L.S.H.B.C.S.H. A Survey on Segment Anything Model (SAM): Vision Foundation Model Meets Prompt Engineering, 2024, [arXiv:cs.CV/2306.06211].
145. Traub, M.; Butz, M.V. Rethinking Vision Transformer for Object Centric Foundation Models, 2025, [arXiv:cs.CV/2502.02763].
146. Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging Properties in Self-Supervised Vision Transformers, 2021, [arXiv:cs.CV/2104.14294].
147. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. DINOv2: Learning Robust Visual Features without Supervision, 2024, [arXiv:cs.CV/2304.07193].
148. Jiang, D.; Liu, Y.; Liu, S.; Zhao, J.; Zhang, H.; Gao, Z.; Zhang, X.; Li, J.; Xiong, H. From CLIP to DINO: Visual Encoders Shout in Multi-modal Large Language Models, 2024, [arXiv:cs.CV/2103.08825].
149. Abobeah, R.; Shoukry, A.; Katto, J. Video Alignment Using Bi-Directional Attention Flow in a Multi-Stage Learning Model. *IEEE Access* **2020**, *8*, 18097–18109.
150. Gautier, T.; Le Guernic, P.; Besnard, L.; Talpin, J.P. The polychronous model of computation and Kahn process networks. *Science of Computer Programming* **2023**, *228*, 102958.

doi:10.20944/preprints202501.0521.v3

151. Becker, P.H.E.; Arnau, J.M.; González, A. Demystifying Power and Performance Bottlenecks in Autonomous Driving Systems. In Proceedings of the 2020 IEEE International Symposium on Workload Characterization (IISWC), 2020, pp. 205–215.
152. Schuman, C.D.; Kulkarni, S.R.; Parsa, M.; Mitchell, J.P.; Date, P.; Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science* **2022**, *2*, 10–19.
153. Deng, L.; Roy, K.; Tang, H. Understanding and Bridging the Gap between Neuromorphic Computing and Machine Learning. *Frontiers Media SA.* **2021**.
154. Zheng, X.; Liu, Y.; Lu, Y.; Hua, T.; Pan, T.; Zhang, W.; Tao, D.; Wang, L. Deep Learning for Event-based Vision: A Comprehensive Survey and Benchmarks, 2024, [arXiv:cs.CV/2302.08890].
155. Benyahya, M.; Collen, A.; Nijdam, N.A. Analyses on standards and regulations for connected and automated vehicles: Identifying the certifications roadmap. *Transportation Engineering* **2023**, *14*, 100205.
156. Wolford, B. What is GDPR, the EU's new data protection law? https://gdpr.eu/what-is-gdpr/. Accessed: 2025-01-09.
157. Wang, X.; Zhu, W. Advances in neural architecture search. *National Science Review* **2023**, *11*.
158. Avval, S.S.P.; Eskue, N.D.; Groves, R.M.; Yaghoubi, V. Systematic review on neural architecture search. *Artificial Intelligence Review* **2025**, *58*.
159. Chandrala, J. Transfer Learning: Leveraging Pre-trained Models for New Tasks. *International Journal of Research and Analytical Reviews* **2017**, *4*, 809–815.
160. Liu, S. Unified Transfer Learning Models in High-Dimensional Linear Regression. *arXiv* **2024**.
161. He, Z.; Sun, Y.; Liu, J.; Li, R. TransFusion: Covariate-Shift Robust Transfer Learning for High-Dimensional Regression. *arXiv* **2024**.
162. He, Z., S.Y.L.J.; Li, R. AdaTrans: Feature-wise and Sample-wise Adaptive Transfer Learning for High-dimensional Regression. *arXiv* **2024**.
163. Lee, J.H.; Kvinge, H.J.; Howland, S.; New, Z.; Buckheit, J.; Phillips, L.A.; Skomski, E.; Hibler, J.; Corley, C.D.; Hodas, N.O. Adaptive Transfer Learning: A simple but effective transfer learning, 2021.
164. Guo, W.; Zhuang, F.; Zhang, X.; Tong, Y.; Dong, J. A comprehensive survey of federated transfer learning: challenges, methods and applications. *Frontiers of Computer Science* **2024**, *18*.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.