


## Review

# Artificial Intelligence (AI) and Machine Learning for Multimedia and Edge Information Processing

Jasmine Kah Phooi Seng <sup>1,2,\*</sup>, Kenneth Li-minn Ang <sup>3</sup>, Eno Peter <sup>4</sup> and Anthony Mmonyi <sup>5</sup> <sup>1</sup> School of AI & Advanced Computing, Xi'an Jiaotong Liverpool University, Suzhou 215123, China<sup>2</sup> School of Computer Science, Queensland University of Technology, Brisbane, QLD 4000, Australia<sup>3</sup> School of Science and Engineering, University of the Sunshine Coast, Petrie, QLD 4502, Australia; lang@usc.edu.au<sup>4</sup> Department of Computer Science, Federal University, Oye-Ekiti 370112, Nigeria; eno.peter@fuoye.edu.ng<sup>5</sup> Department of Electrical and Computer Engineering, Afe Babalola University, Ado-Ekiti 360102, Nigeria; mmonyica@abuad.edu.ng

\* Correspondence: kahphooi.seng@qut.edu.au

**Abstract:** The advancements and progress in artificial intelligence (AI) and machine learning, and the numerous availabilities of mobile devices and Internet technologies together with the growing focus on multimedia data sources and information processing have led to the emergence of new paradigms for multimedia and edge AI information processing, particularly for urban and smart city environments. Compared to cloud information processing approaches where the data are collected and sent to a centralized server for information processing, the edge information processing paradigm distributes the tasks to multiple devices which are close to the data source. Edge information processing techniques and approaches are well suited to match current technologies for Internet of Things (IoT) and autonomous systems, although there are many challenges which remain to be addressed. The motivation of this paper was to survey these new paradigms for multimedia and edge information processing from several technological perspectives including: (1) multimedia analytics on the edge empowered by AI; (2) multimedia streaming on the intelligent edge; (3) multimedia edge caching and AI; (4) multimedia services for edge AI; and (5) hardware and devices for multimedia on edge intelligence. The review covers a wide spectrum of enabling technologies for AI and machine learning for multimedia and edge information processing.

**Keywords:** multimedia processing; edge multimedia; intelligence edge; edge AI; edge computing; edge multimedia analytics



**Citation:** Seng, J.K.P.; Ang, K.L.-m.; Peter, E.; Mmonyi, A. Artificial Intelligence (AI) and Machine Learning for Multimedia and Edge Information Processing. *Electronics* **2022**, *11*, 2239. <https://doi.org/10.3390/electronics11142239>

Academic Editor: Dimitris Kanellopoulos

Received: 15 June 2022

Accepted: 6 July 2022

Published: 18 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The advancements and progress in artificial intelligence (AI) and machine learning [1,2], and the numerous availabilities of mobile devices and Internet technologies together with the growing focus on multimedia data sources [3] and information processing techniques [4] have led to the emergence of new paradigms for multimedia and edge information processing, particularly for urban and smart city environments [5–7]. These paradigms are driven by the convergence of various trends including: (1) the pervasiveness of the deployment of Internet of Things (IoT) sensors and devices in urban areas and smart cities; (2) the emergence of autonomous systems; and (3) the increasing need for multimedia devices and information processing from embedded cameras and mobile imaging devices (e.g., drones). Compared to cloud processing approaches where the data are collected and sent to a centralized server for information processing, the edge information processing paradigm distributes the tasks to multiple devices which are close to the data source.

Edge information processing techniques and approaches are well suited to match the current technologies for Internet of Things (IoT) and autonomous systems, although there are many challenges which remain to be comprehensively addressed. In conventional

computing architectures, the data collected from IoT sensors are converted to digital representation and then sent to the cloud for information processing. This conventional approach has a significant disadvantage which can lead to a decreased performance in terms of latency, energy consumption and communication bandwidth. Edge computing devices can perform the required information and signal processing tasks at the edge of networks [8]. This approach has several advantages: it can reduce the amount of data for transmission and is a promising solution to address the challenges for latency and energy/power consumption.

Multimedia is a combination of text, audio, images, video or animation to form a single interactive presentation. Multimedia processing is the application of signal processing tools to multimedia data to allow the representation, interpretation, encoding and decoding of data. The goals of multimedia processing are the access, manipulation, exchange and storage of multimedia content. Multimedia processing is faced with the challenges of combining video information with sound, text and image into a single communication channel. Another trend is that Internet data traffic is increasingly moving towards multimedia data sources, and sensors/cameras are producing high volumes of multimedia data particularly in urban and smart city environments to address ever smarter applications and services for mitigating and reducing traffic congestion and detecting roadside accidents and hazards, surveillance and security applications and pedestrian detection for autonomous vehicles [9]. New and advanced techniques, approaches and tools are required to be developed to be able to extract the valuable data and insights which can be obtained from the large volume of multimedia data. The recent advancements and progress in relation to intelligent algorithms, combined with powerful computational capabilities and big data approaches have achieved success in using data for analysis, prediction and decision making. The large amount of data generated by IoT and multimedia sources have the potential for enabling AI models to be deployed into smart devices for intelligent analysis and decision making.

Edge computing frameworks have the advantages of being able to efficiently offload large-scale multimedia information processing. There are various ways in which multimedia data modalities can be implemented and deployed on edge devices. These could range from speech and images to video and text data. The benefits of edge computing can be summarized as follows: (1) Improved computational processing and energy efficiency—compared to computations using centralized cloud servers, edge computing distributes the computation tasks to multiple devices (e.g., IoT devices) where each device contributes to a portion of the required computation task. The distribution of the computation tasks can also be configured to take into account the available energy resources for the IoT devices. (2) Reduced data latency—due to the edge devices being located close to the multimedia data sources, the requirements for network transmission can be reduced. (3) Improved privacy and security—compared to cloud servers, edge servers have two significant advantages for privacy and security. The first advantage is that the edge servers are distributed making it more difficult for adversaries to mount a successful attack and the second advantage is that the distributed nature of the edge devices enables the easier monitoring and protection of sensitive information.

Signal processing is a research area in electrical and electronic engineering which focuses on analysis, synthesis and signal transformations. The signals may have different modalities such as audio/sound, speech, scientific measurements, images, etc. Signal processing techniques can be used to improve the transmission, storage efficiency and subjective quality and also to emphasize or detect components of interest in a measured signal [10]. Speech processing is the study of speech signals and its processing methods. Aspects of speech processing include the acquisition, manipulation, storage, transfer and output of speech signals. This section describes some works on signal processing with edge information processing. Preprocessing the speech signals can preserve the key features while filtering operations/computations can remove unwanted background noise for information processing.

Two basic filtering operations that are used for filtering signals are the Infinite Impulse Response (IIR) and the Finite Impulse Response (FIR) filters. Another useful operation is signal transformation which converts signals from one domain to another domain. An example of a popular signal transformation is the Discrete Fourier Transform (DFT). Newer signal transformation approaches can utilize compressed sensing (CS) techniques which can operate on signals using a sub-Nyquist sampling rate [10]. Principal Component Analysis (PCA) is another popular approach for signal preprocessing and is often used to perform feature extraction before the classification or regression processes. Optimizing the classification and regression processes is very important as it determines the quantitative and/or qualitative results and the accuracy of the processes. A recent approach for edge computing for speech signals utilized memristors which are nonvolatile memory devices that have capabilities for in-memory computing. The authors in [11] reviewed the recent progress on memristor-based signal processing methods for edge computing, especially on the aspects of signal preprocessing and feature extraction. The process for the signal filtering operation is based on the convolution operation and can be accelerated with memristor-based computations. The authors in [12] proposed a design for an Infinite Impulse Response (IIR) filter with memristor arrays. Other than filtering operations, more advanced approaches combining signal processing and machine learning (e.g., SVM, random forest, Bayesian approaches, decision tree, etc.) have also been developed. The authors in [13] offered a comprehensive discussion on the application of signal processing and machine learning for intelligent sensor networks. In this work, the authors discussed advanced signal processing approaches including compressive sensing and sampling, approaches using distributed signal processing and intelligent signal learning.

This paper aims to survey these new paradigms for AI and machine/deep learning for multimedia and edge information processing from several perspectives. The paper will discuss how edge and IoT platforms can be effectively utilized to meet the challenges for multimedia information processing in distributed environments. For ease of discussion, we have identified and categorized these challenges as being related to: (1) multimedia analytics on the edge empowered by AI; (2) multimedia streaming on the intelligent edge; (3) multimedia edge caching and AI; (4) multimedia services for edge AI; and (5) hardware and devices for multimedia on edge intelligence. The paper concludes with some application use cases and recommendations for the practical deployment of multimedia information processing for smart city environments in the areas of deployment of the intelligent edge for surveillance and monitoring, human computer interaction (HCI) and health.

There are some other surveys and research works which have been performed for multimedia, edge computation and/or AI technologies. However, these works do not focus on the integration of these technologies to form a new paradigm. The motivation of this paper is to address this gap in the earlier works. Compared to other and earlier works, this paper makes comprehensive contributions towards the multimedia edge AI/ML paradigm from different aspects:

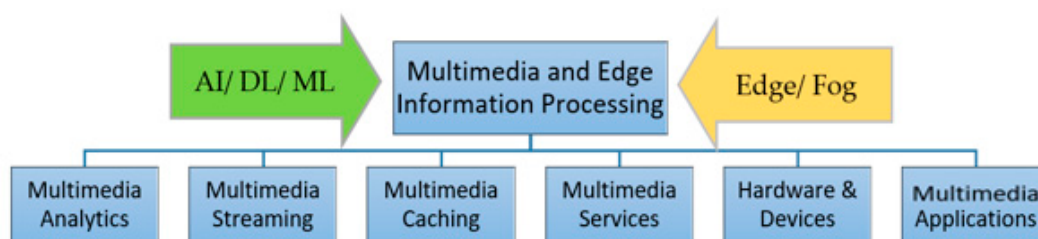
- The review covers a wide spectrum of enabling technologies for multimedia processing and edge AI. Different types of architectures, approaches and techniques for multimedia and edge AI/ML are discussed. We also include new paradigms covering models and architectures for multimedia streaming, services and caching for edge AI;
- The review covers the emerging paradigms of the deployment of hardware and devices for multimedia on edge intelligence, and the role of the IoT in edge AI systems. The final part of the paper gives recommendations for future work in multimedia and edge AI;
- Use cases for smart city environments in the areas of deployment of the intelligent edge for surveillance and monitoring, human computer interaction (HCI) and health are used to illustrate the benefits and potential of edge AI/ML systems;
- To aid researchers and designers, the paper gives a roadmap (Table 1) showing an overview of the classification descriptors which are covered in this paper. This can

serve as a comprehensive reference and point researchers and readers towards further and future works.

**Table 1.** Classification descriptors and areas covered in the paper.

Classification Descriptor	References
Background information on the edge paradigm, multimedia and AI	
Models and Technologies: Fog, Mobile Edge Computing and Cloudlet	[14–16]
AI Architecture Models	[17–23]
AI Learning Models	[24–30]
Multimedia Technologies and Hardware Devices on Edge AI	
Multimedia Analytics on the Edge Empowered by AI	[31–48]
Multimedia Streaming on the Intelligence Edge	[49–55]
Multimedia Caching for Edge AI	[56–73]
Multimedia Services for Edge AI	[74–77]
Hardware and Devices for Multimedia on Edge Intelligence	[78–91]
Representative Use Cases/Applications of the Multimedia Edge Paradigm	
Use Case 1: Intelligent Multimedia Processing on the Edge for Surveillance and Monitoring	[92–98]
Use Case 2: Intelligent Multimedia Processing on the Edge for Human Computer Interaction (HCI) and Health	[99–111]

The remainder of the paper is structured as follows: Section 2 provides initial discussions and background information for the edge paradigm, multimedia information processing and artificial intelligence. The next five sections provide in-depth discussions on five areas of interest. Section 3 provides discussions on multimedia analytics on edge empowered by AI. Section 4 provides discussions on multimedia streaming on intelligent edge. Section 5 provides discussions on multimedia edge caching and AI. Section 6 provides discussions on multimedia services for edge AI. Section 7 provides discussions on hardware and devices for multimedia on edge intelligence. The next two sections (Sections 8 and 9) provide discussions on representative use cases focused on applications for the multimedia edge paradigm. The end of Section 9 gives some recommendations for future work in multimedia and edge AI/ML. The paper is concluded in Section 10. Figure 1 shows a roadmap and a summary of the research areas covered in the paper and Table 1 shows the classification descriptors and areas covered in the paper.



**Figure 1.** Summary of the research areas covered in the paper.

## 2. Background Information

This section offers background information and an overview of the edge paradigm and its related concepts and models towards AI before further detailed discussions are provided in the later sections of the paper. The section covers the following areas: (1) fog, mobile edge computing (MEC) and cloudlet models and technologies; (2) AI architectures; and (3) AI learning models.

### 2.1. Models and Technologies: Fog, Mobile Edge Computing (MEC) and Cloudlet

The fog model or what is termed as fog computing deploys end devices known as fog nodes at the edge of the network or gateway to perform distributed information processing. Fog computing can be deployed to collaborate with IoT nodes and perform the required analytics in a distributed manner [14]. Fog computing has the advantage of reducing the data latency and provides real-time collaborative services. Mobile edge computing or what is termed as MEC is used on radio access networks (RANs) and cellular networks to provide computation and storage services at the edge of the networks. Similar to the fog model, the MEC model is also a distributed approach. In recent years, MEC has been closely linked and associated with 5G network technologies to provide services for autonomous vehicles and wearable computing platforms [15]. The cloudlet model deploys small clusters with computation and storage capabilities to assist the information processing requirements for mobile devices and smartphones [16]. These clusters can be deployed in areas such as shopping centers and building environments. The cloudlet can be seen as the middle layer to link mobile devices to the central cloud and provides services to assist the mobile applications.

### 2.2. AI Architecture Models

There are different AI architecture models which can be deployed for multimedia on the edge. This sub-section gives descriptions for several architectures which have been proposed such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), long short-term memory (LSTM), restricted Boltzmann machines (RBMs) and autoencoders.

Convolutional neural networks (CNNs) [17] are well-known and established deep neural network (DNN) AI architectures which have been designed to process multidimensional and spatial data such as images. CNNs are often used in multimedia applications such as computer vision, object detection and classification. A CNN architecture is composed of several convolutional layers which are used to extract the spatial information and correlations in the data. Recurrent neural networks (RNNs) [18] are designed to process sequential or time series data. The RNN architecture takes inputs from the current and previous samples and the stored state from previous time steps to remember previous trends in the time series data. RNNs are often used in multimedia applications such as activity recognition and speech or language processing. The long short-term memory (LSTM) AI model [19] can be considered as an extension of the RNN model. The LSTM model uses a neuron structure termed as a memory cell which includes a multiplicative forget gate, input gate and output gate which are used to control the access to the memory cells.

Generative adversarial networks (GANs) [20] have an AI architecture which consists of two neural network architectures which are termed as the generator network and the discriminator network. The generator and discriminator networks work in tandem to find an optimal solution to the problem being investigated. The aim of the generator network is to produce new data after it has learnt the distribution of the data, whereas the aim of the discriminator network is to distinguish if the input data are coming from the generator network or the real data stream. A useful application for GANs is to create more data samples for training AI architecture models (i.e., data augmentation).

The restricted Boltzmann machine (RBM) [21] is a stochastic neural network architecture with a probabilistic graphical model. The typical RBM architecture has two layers (input and hidden layers) that contain the latent variables. RBMs are often used in multimedia applications such as collaborative filtering and anomaly detection. The autoencoder (AE) [22] is another AI architecture which includes an input layer, hidden layers and an output layer. The AE consists of two parts: (1) an encoder component which learns the representative characteristics of the input in a compressed form and (2) a decoder component which aims to reconstruct the original input data from the compressed form while minimizing the reconstruction error. There are several recent AI architectures which have shown promise for multimedia AI such as the transformer architecture and the variational



auto-encoder (VAE) architecture. The transformer architecture can be used with CNNs for applications involving object detection and localization. The VAE is a self-supervised network architecture which consists of an encoder and a decoder network. The encoder network performs the mapping of the image into a latent code space and the decoder network performs the image generation from a latent code. Further information on recent AI architectures can be found in [23].

### 2.3. AI Learning Models

There are different AI learning models which can be deployed for multimedia on the edge. This sub-section provides descriptions of several learning models which have been proposed such as deep learning, reinforcement learning, deep reinforcement learning, federated learning and transfer learning. Deep learning models focus on creating large neural network models that are capable of making accurate data-driven decisions [24]. This type of AI learning approach is particularly suitable for training complex data when there are large datasets available. In cognitive science, reinforcement learning is a learning model which is designed for scenarios in which an agent interacts with an environment that provides rewards and/or punishments [25]. Deep reinforcement learning (DRL) [26] combines deep learning and reinforcement learning with the objective to build intelligent agents from large datasets which are able to determine the best actions to perform for a various set of states by interaction with the environment. The DRL achieves this by maximizing the long-term accumulated rewards. DRL approaches are often used in multimedia applications for resource allocation (e.g., determining the optimal rate for video transmission) and recommendation. There are two general categories for DRL, which are value-based models and policy-gradient-based models.

Transfer learning (TL) [27] approaches can be used to reduce the training costs for AI architectures on edge devices. In this approach, a base network (teacher network) is first trained. The learned features are then transferred to a target network (student network) for use in training a target data set. The authors in [28,29] performed various studies to quantify the performance gains of transfer learning for accuracy and speed of convergence. Federated learning (FL) [30] is a decentralized learning or training approach which can be utilized to achieve privacy for the edge devices and information processing. In this approach, the training data from edge devices are not sent and aggregated on a centralized data center. Each edge or mobile device performs a distributed training approach and trains a shared model on the server by performing the aggregation of locally computed updates. Contrastive learning [23] is a recent AI learning model which assigns an energy level to training examples of a video and a possible continuation. The objective of this learning model is to give a degree of goodness or badness to the training examples for which no or few labels are used. This approach trains the neural network to produce similar output vectors for different views of the same object, and different output vectors for views of different objects.

## 3. Multimedia Analytics on the Edge Empowered by AI

This section discusses the concept of multimedia analytics on the edge empowered by Artificial Intelligence. The multimedia data generated by most IoT devices are very large and are faced with challenges related to a limited bandwidth when transferred to the cloud. Edge computing is an important tool in overcoming the challenges related to cloud computing such as network congestion, delay in response time, cost, etc. This edge computing helps to introduce the idea of processing data and decision making in the network. To access this data, artificial intelligence and machine learning is used.

Qu, et al. [31] proposed a supporting structure called DroneCOCONet, which helps coordinate the processing of large datasets for unmanned aircraft video computational analysis. The coordination of this datasets is made possible with the use of edge computational offloading. The edge computational offloading improves the system performance by freeing the computational workload on the unmanned aircraft and increasing the video

analytics of the drone. The edge computational offloading is made up of two approaches, namely, heuristic-based and reinforcement learning-based approaches. The approaches mentioned above create a quick provision for the coordination of the task for decision making among the multi-unmanned aerial vehicles in a dynamic offloading. In another work, Ilhan et al. [32] applied a pre-trained deep learning algorithm to manage computational offloading in image and video analysis for unmanned aerial vehicles. The system deployment employs artificial intelligence with a view to outperform other existing models in energy conservation and the utilization of remote edge caches operating on LTE servers for bandwidth optimization.

The computational offloading delivery captured in the authors' research work was expressed as two problem sets modelled using the heuristic decision-making process and the Markov decision process. The aim of their research work was to reduce the expenses involved in the computation and delay in the edge resources. The authors introduced a learning-based dynamic computational offloading and control networking through DroneCoCoNet that aids artificial intelligence-backed reinforcement learning-based processing in the edge computation offloading plan and also a resource-aware network protocol selection based on application requirements. An overview of the architectural structure of the DroneCoCoNet system is shown in Figure 2. This architectural structure illustrates how the DroneCoCoNet system is embedded in a multi-drone-edge-server situation. This includes the computation offloading between the drone and ground control station connected to the edge server and the required communications. In the setup of the multi-drone communication, a hierarchical Flying Ad-Hoc Network (FANETs) is considered with properties ranging from a low capability search to high capability smart gathering drone systems.

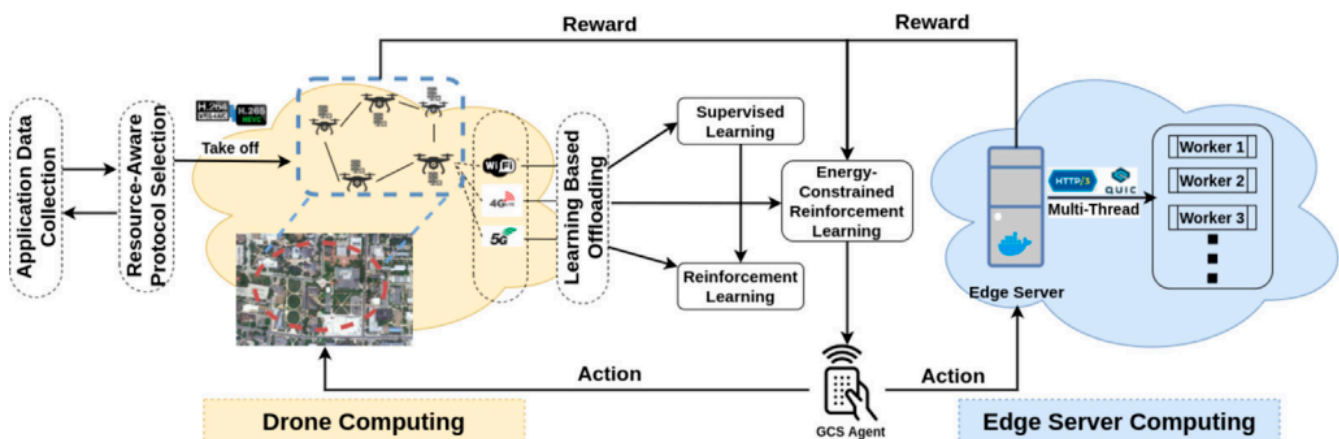


Figure 2. The architectural structure of DroneCoCoNet [31] (Qu, et al., 2021).

In the hierarchical FANET method, three major logical modules are involved, which include an application-level data collection component, a resource-aware protocol selection module and the intelligent offloading module. The final part of the logical modules represents the edge computation offloading and the rest is the control layer of the network. The novel learning-based computation offloading algorithm is used to increase the utilization of the resource and the performance of the system such as the energy consumption, analytics time and accurate processing of the data. The authors addressed the issue associated with choosing the network protocol and drone-edge computational task offloading. Moreover, to ensure that the drone video analytic system can achieve more stability and reliability, the authors addressed the issues involved in the selection of the network protocol and the drone-edge computational task offloading. The authors introduced two significant learning-based methods involving supervised and reinforcement learning-based algorithms. In the discussions surrounding computational offloading, the application of the machine learning-based forecast and task scheduling are common themes required for optimized performance. The research further introduced a novel application called QUICer to en-

hance the parallel transmission of data capable of serving multiple clients or server side user requirements. QUICer applies HTTP/3 operating over the QUIC transport protocol establishing quick data transmission capabilities.

Edge computing is very important in overcoming the challenges related to cloud computing such as network congestion, delay in response time, cost, etc. It helps to introduce the idea of processing data and decision making in the network. Monburinon et al., (2019) [33] proposed a work focused on the use of Internet of Things, deep learning and an edge computing scheme for image recognition in agriculture. A recognition system called hierarchical edge computing was used to recognize the presence of animals and to determine the types of animals on the farm. Its major processing involved a low-cost gateway system such as Raspberry Pi. A convolutional neural network dynamic which is a learning method was introduced to identify or classify some images tasks in the network. The authors developed a framework called Deployment Environment Aware Learning (DEAL) for the deployment of image recognition. In their work, an engine which is used for recognition on the edge server reduces latency in the network. The image recognition is made up of the three layers. First, the physical layer helps to handle tasks such as the acquisition of data, detection of movement and the capturing of animal images on the farm. Second, the edge computing layer performs the computational tasks such as image and data processing, and animal recognition. It also helps to store information for a short term and acts as a link between the physical layer and the cloud computing layer. In the agricultural environment, the deployment of an edge server at different points in the environment helps in the detection of animals. Individual servers perform the detection of different animals independently. Third, the cloud computing layer performs difficult tasks involving Convolutional Neural Network training and the analysis of data. This third layer maintains a large store of training data that needs to be preprocessed. An example of a server that belongs to this layer is the high-performance cloud server. However, the exchange of information takes place between the edge computing layer and the cloud computing layer.

Munir et al., (2021) [34] introduced the concept of data fusion and Artificial Intelligence at the edge to carry out an intelligent task. The fusion helps to give accurate and unambiguous data to the Artificial Intelligence. A comparative study was made on various data fusion and Artificial Intelligence patterns. The authors expounded the different stages of fusion and its relationship with the various types of Artificial Intelligence. There are three levels involved in the examination of artificial intelligence: firstly the edge-of-network sensor, or the Internet of Things nodes, secondly, edge servers and thirdly the cloud servers. The term “Edge Artificial Intelligence” refers to Artificial Intelligence at the edge of the network sensor and edge servers. The edge Artificial Intelligence minimizes the transmission challenges on the network and allows computing close to the edge of the network. Data fusion is carried out at each level of the architectural structure to reduce redundancy from the primary data. The experimental results showed that artificial intelligence with data fusion improves the execution of tasks, gives a higher accuracy and reduces energy consumption compared to Artificial Intelligence without data fusion. The alignments of various levels of fusion and Artificial Intelligence are categorized by using the seven levels of the Data Fusion Information Group (DFIG) model. Level 0 in the DFIG model is called the low-level data fusion, level 1 is intermediate-level data fusion and level 2 to level 6 in the DFIG model represent the high-level data fusion.

Due to network congestion during transmission, Kim et al., 2021 [35] proposed a modular structural design with deep neural networks. This proposed structure gives an answer to the problem in actual video analysis in the edge-computing platform. The deep neural networks have two parts called the Front and Back Convolutional Neural Networks. The Front Convolutional Neural Network utilizes a Shallow three-dimensional Convolution Neural Network (S3D) and the Back Convolutional Neural Network utilizes a two-dimensional Convolution Neural Network (2D). The Front Convolutional Neural Network is used to reduce the size of many video frames by combining video frames into a single unit which



is constituted of three channels containing feature maps (3CFM). A pre-trained 2D CNN is used because of its compatibility with two-dimensional imaging exhibiting Red, Green and Blue (RGB) channels. The 2D CNN is modified by three subsampled grayscale images obtained from video frames. These eventually form a Stacked Grayscale 3-channel Image (SG3I) capable of the same level of compatibility as the initial system.

Jainuddin et al., (2020) [36] applied deep neural networks (DNN) to the performance classification of Google's machine learning hardware (Edge TPU). This was achieved by subjecting samples of images to different categories of deep learning models. The application projected the advantages by dedicating the full processing capacity of the CPU to DNN models. Chaitra et al., (2021) [37] proposed a deep learning model for the recognition of plant disease on edge devices utilizing plant images. With just a click of the leaf picture, plant diseases can be diagnosed, and possible solutions highlighted. Two methods of deep learning are applied: the first involves on-premise mobile devices without an internet connection and the second operates over the web service creating the need for network services. The latter option creates the need for high accuracy and database scalability to manage the addition of plant disease data over long periods of time. In the case of no network connection on a mobile, a tflite model is used. The tflite model was proposed as an option offering low storage capacity requirements and an optimized performance. The proposed model has the disadvantage of a poor performance when applied on a data set type that has not been seen by the model.

Video analytics are faced with many challenges such as scheduling on mobile devices due to the differences in memory requirement and the time required for the processing of multiple CNN models. In another methodology, Tan and Cao, 2021 [38] introduced resource-aware scheduling algorithms to tackle similar challenges. Their resource-aware scheduling algorithm was implemented on Android-based smartphones which are characterized by joint offloading and local processing methods to speed up video processing time. In mobile communication, the transmission of short videos to an existing network is identified as a barrier to the users' QoE. Deep reinforcement learning for video quality selection on MEC platforms utilizing Radio Bearer Controls (RBC) was applied by Wu et al., 2019 [39] and Chen et al., (2020) [40]. Wu et al. (2019) proposed a use case relevant for content caching in short video applications, creating duplicates of short video content for quick playback. Chen et al., (2020) proposed a multi-user edge assisted video analytics (MEVAO) methodology to perform the functions of task offloading. The above problems are both expressed as Markov decision processes, with the objective of exploring opportunities for network policy training, which establishes a long-term performance in video quality rendering. The identified benefits included quality-related profits and the reduction in the cost posed to bearers in terms of penalties for latency.

Ran et al., (2018) [41] presented a mobile deep learning framework for edge video analytics that combined computationally weak front-end devices (i.e., smartphones) with higher performing backend devices with a greater processing capacity. The combination creates an opportunity for the execution of deep learning processes to happen either at a local or remote level. Further to this, the convolutional neural network (CNN) is deployed for the detection of real-time objects for augmented reality applications. The performance in terms of optimal offloading strategy model accuracy is measured in terms of quality of video playbacks, rates of battery depletion, bandwidth performance and network delays. Convolutional neural networks were also referenced by Peng et al., 2021 [42] in the development of an intelligent video analysis capable of providing security functions for the power industry. CNN deployment in this use case will support intrusion and pyrotechnic detection utilizing edge nodes and cloud training data sources.

Deep learning models require a large number of computational resources when applied to large-scale Internet of Things data. There are problems with the network when connecting the source of data capture with the cloud platform. In a paper titled "Edge Enhanced Deep Learning System for Large-scale Video Stream Analytics", Ali et al., 2018 [43] identified the opportunity to manage the problem with bottlenecks resulting from the

transfer of a large volume of multimedia data to the cloud networks. The author proposed a solution which involves channeling deep learning data sources through edge and cloudlet resources to reduce latency and bandwidth costs and to improve performance. The proposed method eliminates the heavy dependence on cloud resources and executes initial data processing at edge and fog nodes. The proposed method architecture is made up of three logic-based processes taking place at the edge and cloudlets from which the streamed data flows to the final logic layer located on the cloud server. The tiering system represented does not exhibit any dimensional constraints with the possibility for vertical and horizontal tiering systems. The Vertical Tier Scalability (VTS), which relates to the local area network (LAN) exhibits single or multi-computational node configurations existing as clusters. In Horizontal Tier Scalability (HTS), sequential processing takes place along nodes of a similar capacity at the interface with VTS clusters existing at some physical distance. The resulting system exhibits the combined properties of VTS parallel video processing and HTS sequential video processing within a single logical tier.

Tsakanikas and Dagiuklas (2021) [44] presented a novel distributed Artificial Intelligence Edge Video model that enables real-time processing. The proposed model is built on Virtual Function Chaining (VFC) applicable for distribution of Artificial Intelligence applications across an edge network. The proposed work developed a method that defines the optimum design for a VFC, a system architecture which is hosted on the Virtual Function Orchestrator (VFO), a framework that provides the optimized sharing of AI video surveillance services to edge-based resources. The research evaluated a prototype of the framework applied for the evaluation of model performance. The set performance measures were directed at establishing the feasibility and level of effectiveness of the VFCs in real world AI surveillance applications. The established model benefits from caching to achieve the level of scalability required for the volumes of data characteristic of big data applications.

Zhou et al., (2021) [45] introduced “Flow Edge-based Motion-Attentive Network (FEM-Net)” which addresses the issues related to unsupervised video object segmentation (UVOS). UVOS has been identified as a challenge in applications involving functions such as object identification, security and video compression. The Flow Edge-based Motion-Attentive Network can effectively separate the moving objects using dedicated flow-edge video tools. Most of the research on flow-based methods has only considered the direct method using the optical flow and this can result in the models being misled to disjoint the foreground objects incorrectly. Moreover, using the edge-blur optical flow can result in the false detection of segmentation boundaries. The FEM-Net has two stages: First, the flow edge connect stage which involves the use of the Flow Edge Connect module (FEC). The FEC applies a lower elaborate motion and fewer appearance features to connect the poorly represented portions of video frames. Second, the edge-based object segmentation mask synthesis module is responsible for the generation of segmentation masking for centralized objects.

In optical remote sensing for harbor monitoring, there are two major research fields covering the areas of sea-land segmentation and ship detection. Cheng et al., 2017 [46] combined these two major research fields into a single structure with the help of FusionNet and utilized deep convolutional neural networks in the forecast of pixel-level labels for inputs. This ability to provide this feature has been found to offer benefits related to the early discovery of semantic segmentation challenges. The proposed solution applies an edge-aware convolutional neural network which provides remote sensing harbor images with a focus on three distinctive objects: sea, land and ship. The challenges with harbor image processing can be grouped into three. The results from imaging are influenced by complex inconsistencies in the land texture which influence sea and land segmentation and ship detection. There is also the challenge of drawing a clearly defined segment between sea and land boundaries. Third is the concern of the distortions that clouds, waves and shadows pose to the results of image segmentation. The authors carried out the research by designing a multi-task model which trains the segmentation and edge detection networks

simultaneously. The segmentation network provides pixels with allocated class labels while the edge network is responsible for determining class boundaries. The semantic features obtained from the segmentation network are used to determine the placement of edge network. Multitasking takes place at the edge layer while the encoder and decoder models are utilized as the elementary segmentation network.

The edge-aware convolutional network is comprised of two parts: the encoder and the decoder network. The encoding network component is made up of several convolution and pooling operations with the objective of feature extraction. The decoding network component performs a symmetrical operation which covers four areas which include unpooling layers and its corresponding convolution, a Batch Normalization (BN) layer and Rectified Linear Units (ReLU) layer. The latter two layers are used to speed up the learning process within the network. A SegNet is used as the elementary structure of the segmentation network based on its ability to segment with greater detailing. The SegNet model provides a solution to the problem associated with the loss of feature maps information. The SegNet model solution to the highlighted problem involves loading a decoder component within the sections required for encoding.

In autonomous vehicles, video analysis is very important in improving safety. The large amount of video data poses challenges in the autonomous vehicle networks. The instability in the network connection causes a lack of security in data sharing. Jiang et al., (2020) [47] introduced a method for video resource allocation based on a machine learning-backed model that leverages blockchain technology and edge computing for Internet of Autonomous vehicles (IoAV). The integration enables the optimization of the blockchain system operation throughput and the minimization of the multi-access edge computing system delay. The joint optimization problem in deep reinforcement learning is expressed in the form of a Markov decision process (MDP) and an asynchronous advantage actor-critic (A3C) algorithm which is capable of learning with each interaction.

Kristiani et al., (2020) [48] introduced a deep learning model capable of optimizing image classification on edge networks. The utilization of image preprocessing and data augmenting schemes enables the structuring of data for the learning process. The CPU optimization and hyper parameter tuning are used to speed up the deep learning training process. The following represent the topology used in their deep learning evaluation: InceptionV3, VGG16 and MobileNet. The utilization of InceptionV3 enables the modeling of deep learning applications while the model optimizer enables the optimization of the trained model on the edge. Training processes were performed on cloud resources while deduction processes were executed on the edge network. The experiments conducted showed that mobilenet had the lowest accurate model and the most time required for model deployment when compared to VGG16 and InceptionV3. Moreover, VGG16 had the most dependable and the least time required to load the model.

#### 4. Multimedia Streaming on Intelligence Edge

Streaming involves the continuous transmission of multimedia files in bit sized flows from client servers to users allowing content consumption without the need to establish permanent storage spaces for transmitted data. Video streaming has become a major source of internet traffic generation with major demand placed on the edge network infrastructure to provide capacitive storage capable of managing upload and download operations for an ever-increasing number of users. Additionally, there is a growing presence of IoT devices operating over sensor networks, constantly transmitting multimedia data obtained from sensor nodes designed to capture varying physical, chemical and statistical properties. According to data obtained from Cisco visual networking index for 2016 to 2021, an estimated 41 exabytes of data were transmitted on a monthly basis at initial measurements and this was projected to increase to 77 exabytes by 2022 with between 79 and 82% of this traffic made up of video data. This highlights the critical role played by multimedia streaming on bandwidth efficiency in mobile edge computing. This section examines the work relating to the application of intelligent solutions for the optimized streaming

of multimedia content over content distribution networks and similar platforms within edge networks.

The streaming of videos across the edge infrastructure demands large bandwidth allocations. Some service providers manage this by establishing fair usage policies which manage the user experience reactively based on historic data. Considerations for real-time bandwidth sharing to optimize the allocation across a section of users with video quality adaptations were explored by Chang et al., 2019 [49]. The research applied a Deep Q-learning approach to inform a bandwidth sharing policy operated within an edge network simulation. The MEC server utilized in the experiments was simulated using the LTE mobile cellular network software based on Amarisoft EPC Suite and eNB as representative edge nodes operating on separate physical machines. The experiment established two scenarios constituting the experience of a single user and two users modelled to observe how bandwidth allocation will be executed with respect to quality of experience and fairness of allocation among users. The information from the MECs Radio Network Information system was extracted to establish quality of experience metrics for performance evaluation.

The authors adopted Deep Q-learning as an alternative to standard Q-learning to overcome the requirement for large training data sizes, since standard Dynamic Adaptive Streaming over HTTP (MPEG-DASH) only specifies media presentation formats and creates flexibility for adaptation logic. Deep Q-learning provides a more adaptive option for video fragmentation than previous comparatively rigid client-based logic algorithms. Compared to previously applied client-based logic used to coordinate HTTP Adaptive streaming (HAS), Deep Q-learning leverages the neural network to establish reliable data sets similar to Q-tables that provide information on action and rewards over a range of variables which are monitored as performance measures in this methodology. In the research setup, video content unavailable within the internal edge-supported video caches are forwarded to an external video content server. Alternatively, these requests could be redirected to a supporting edge-assisted video adaptation application. These systems are capable of providing users with the multimedia data appropriate to the initial query, while tailoring the user experience to match the video quality based on the limitations imposed by the streaming policy. The research challenge then would be to establish a policy that provides users with the best perceived experiential quality in terms of video quality, time required for downloads and adaptability within the network.

The utilization of the experience replay mechanism in the training process provides the reliability described within existing datasets. This is credited to the development of a heuristic borne of multiple instances within the network. The experience captured includes the initial state, which refers to the network state before any action is taken. The next variable captured is the action which establishes the bitrate for the video segment intended for access and download. The reward variable provides feedback on how effective the action variable was in providing minimum bitrate deviation per user. Finally, a variable is captured to represent the new state of the network. The fairness index measured the bitrate deviation per client. In the two-client scenario, Jain's fairness index is applied which surveys and compares the differences between the adaptation solutions in bitrate delivery. The results from the experiment were considered over two client-based adaptation logic tools: Buffer-Based Adaptation (BBA) and Rate-Based Adaptation (RBA). These were compared against the quality of experience results obtained from Deep Q-learning. For the BBA in the single user scenario, switching buffer rates causes frequent oscillations in bitrate even when download conditions are relatively stable. RBA by design neglects buffer occupancy in providing adaptations leading to a situation where the selected video quality exists as a function of the bandwidth available. The observed high buffer rates are not utilized to optimize the process creating missed opportunities in situations with fluctuating bandwidth. Dash.js creates a high average video quality with a low switching frequency. This creates an experimental situation where a low bandwidth is selected for the entire process even in the presence of a greater available allocation. For the two-user situation dash.js produces a fierce rivalry for bandwidth where one user benefits from



a higher bandwidth allocation and a higher average throughput to the detriment of the competing user.

Zhou et al., (2020) [50] captured the enormous computational demands created by 3D video requirements among an increasing number of users. Their research developed a Quality of Experience Model that relies on actor-critic deep reinforcement learning to adapt video renderings reactively improving video playback buffer times and bandwidth distribution. A resource allocation model (RAM) is hinged on a Software distribution Network (SDN)-managed Mobile Edge computing architecture. The work in [51] discussed several techniques involving AI to manage SDN networks and proposed a SDN management system powered by AI termed as SDNMS-PAI for handling end-to-end networks. The SDN has the function of establishing control of the server resource allocation and allocating resource necessary for data processing requirements separately. The resource allocation model (RAM) is implemented at the edge layer, where 3D video playbacks are cached for onward transmission to users in form of video blocks.

The author identified video blocks as sections of the frame-by-frame video, each accounting for one second of playtime. The choice of video block rates to implement are influenced by the system's need to operate a Dynamic Adaptive Streaming over HTTP (DASH) protocol. Hence, the performance of future video files is reliant on the playback statistics computed. The resource allocation model benefits from caching operations within the MEC along several edge servers which constitute the overall network. Working with the SDN, the authors suggested a method of allocation of MEC resources over the network which is supported by buffers to optimize the 3D video user experience. Caches provide optimized video block transmission while tiling operations which result in stitching video blocks in parts rely on edge computing resources. The quality of experience model (QoEM) is based on an improvement of the resolution allocation to the Head-Mounted Display (HMD) viewport which is responsible for the transmission speeds of 3D videos. The HMD viewport tiles require equal tile rates to mitigate observable screen fragmentation during display. The higher resolution in HMD is complemented by the reduction in the allocation outside the viewport, with tiles in this region allocated a non-zero rate. Allocation rates are modelled using the Markov decision-making Process (MDP) which optimizes quality of experience. An actor-critic deep reinforcement learning tool is deployed at this point to predict and adapt viewports and the bandwidth of future videos. Additional tools applied in this work include Long Short-Term Memory (LSTM) and fully convolutional (FC) networks responsible for providing resolution accuracy. The performance of the methodology was evaluated using model predictive controls (MPCs) and Deep Q-Network (DQN) to perform a comparative analysis on four QoE targets.

Luo et al., (2019) [52] used similar inputs to Zhou et al., 2020 [50] differentiated by the objective to proffer a solution for energy management and quality of user experience in cases where there is a requirement for video streaming over software-defined mobile networks (SDMN) existing on mobile edge computing resources. This is achieved by establishing variables within two optimization problems based on constrained Markov decision process (CMDP) and Markov Decision Process. The optimization problems are solved by applying the model-free deep reinforcement learning; asynchronous advantage actor-critic (A3C) algorithm method. The subsequent analysis and adaptation derived describe video buffer rates, adaptive bitrate (ABR) streaming, edge caching, video transcoding and transmission. The Lyapunov technique was used to address the challenge created by the application of CMDP which applies a one-period drift-plus-penalty. This creates a requirement for the resolution of a period-by-period isolated deterministic problem to create an accurate representation of conditions. The substitution of the one-period drift-plus-penalty with the T-period drift-plus-penalty provides a global solution to the CMDP problem. The streaming profile considers a downlink case involving video transmission within a mobile network with multiple base stations serving a large user demographic. With each request from the edge base station, a discrete time Markov chain (DTMC) is used to model changes in the state of the channel which is dependent on transmission probability. Also important



for the achievement of the anticipated QoE are the buffer rates which perform the duty of smoothing with variations in bitrates. The research approach to buffer sizing is modelled to match demands by mobile devices with the concept of minimum tolerable performance deterioration tolerable at different buffer levels.

The need for indexing video tiles to provide an adaptation based on bandwidth requirement makes a case for the presence of a software-defined controller. The SDN is utilized for QoE adaptation as the bitrate allocation to streamed videos will require constant adjustments to meet the demands of changes in resolution within video content. Beyond quality adaptations, the segmentation of video tiles will be decided by the SDN controller which has the responsibility of assigning computational resources required for transcoding video files from one virtual machine to the final mobile device. In establishing the quality of experience, the author selects performance metrics that measure the time average bitrate which measures the normalized bitrate time average for each segment. Moreover, the time average instability index is measured to depict user perception towards the influence of changing bitrates brought on by intelligent SDN adaptations.

Machine learning in this research served to define an optimal policy that optimizes each scenario of bandwidth demand and allocation utilizing limited learning data. For the simulation of the process, an open-source machine learning library named Pytorch was used to implement the actor-critic deep reinforcement learning. An MDP and a non-MDP optimization solution were proffered to obtain the best performing between the methodologies. The utilization of caches was found to speed up learning rates, whilst the best performing adaptations were found with up to 50 segments cached in a setup that involved 20 base stations with three mobile devices per station. It was further observed that as the number of mobile devices per base station increased, the maximum power consumed at each base station was stretched, leading to major service degradation.

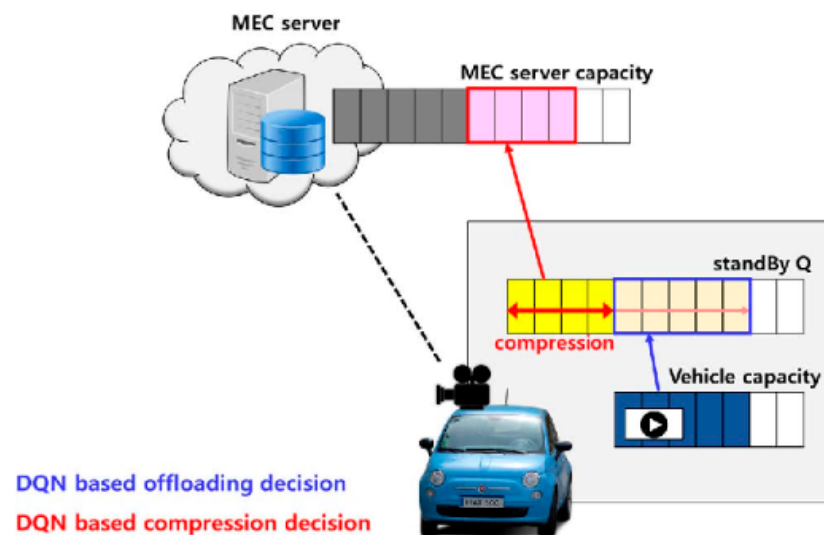
Dai et al., (2021) [53] raised the consideration for multimedia streaming components required for effective communication among vehicles existing within an Internet of Vehicles (IoV) network. The requirement for constant multimedia streams to be maintained with vehicles in constant motion, creating a dynamic demand, differentiated the identified problem from previous considerations. The authors considered streaming for heterogeneous IoV with an adaptive-bitrate-based (ABR) multimedia streaming design which operates over an MEC network. Utilizing roadside units as mount points for edge devices, the bandwidth allocation per vehicle can be determined with priority placed on quality of experience for each user. This means as multimedia streaming takes place, intelligent systems are required to guarantee the quality of the received multimedia segments while minimizing the lags created by bandwidth limitations. Their adaptive-quality-based chunk selection (AQCS) algorithm provides opportunities to monitor and synthesize service quality, playback time and freezing delays within the setup. Other critical quality of experience factors responsible for service performance and freezing delays were synthesized using the joint resource optimization (JRO) problem developed in the paper.

Multimedia segmentation and the allocation of bitrate in their methodology employed Deep Q-learning (DQN) and a multi-armed bandit (MAB) algorithm, both being reinforcement learning-based methodologies. The application of MAB was noted to create a consequent loss of convergence speed and clock speed. Using Q-tables and gradient based Q-function, deep Q-learning is expected to provide better results through rehearsed data driven processes. The multimedia data were hosted on the cloud layer. Multimedia files constituting of varying quality levels to reflect the predictive adaptation of DQN are duplicated at the MEC layer to support ABR. The Multi-Arm Bandit algorithm provides decision making and Q-function updates which are made up of the streaming history and rewards for bitrate adaptations. Follow-up action provided by Deep Q-Learning establishes the representation of system state, experience replay, a loss function and a reward function which leads to a performance index for comparison to other methodologies. Using a traffic simulator alongside a scheduling and optimization module, real-time trace data

were obtained from vehicles within Chengdu city in China over a 16 km<sup>2</sup> area from the resource-based Simulation of Urban Mobility (SUMO).

The author noted the JRO exists in a novel space; hence, a combination of methodologies were utilized to create a suitable comparison. These were comprised of a classical cache algorithm and two adaptive-streaming algorithms made up of a Markov Distribution Process and a Rate Adaptation that will be responsible for chunk transmission. These algorithms were tested with different bandwidth requirements across the multi-arm bandit, DQN, adaptive quality-based chunk selection (AQCS) and least frequently used (LFU) algorithms. Five scenarios were simulated with different traffic conditions observing average and standard deviations for vehicle number and dwelling time. The results captured the effects of traffic workload on the performance of algorithms. Of the five tools considered, the combination of DQN and AQCS was found to perform best in managing average service quality (ASQ) and minimizing the average freezing delay (AFD) simultaneously.

Deep Q-networks find additional applications in managing streaming multimedia data for autonomous vehicles as captured in research presented by (Park et al., 2020) [54]. The research addressed the challenge of establishing reliable video streaming in fast moving autonomous vehicles and proposes a combined Mobile Edge Computing and DQN driven solution. Their design was constituted of two DQN-based decision support applications with one dedicated to the offloading decision algorithm and the other charged with the data compression decision algorithm. Autonomous vehicles benefit from the operation of a large number of digital cameras fitted at differing locations responsible for image capturing and processing. This function has high requirements for speed to support the decision-making processes that influence the safety factor of the vehicles. Caching along the MEC supported by 5G technologies provides reasonable support. However, the method proposed seeks to achieve a greater bandwidth efficiency which promotes video offloading and compression operations for fast streaming as represented in Figure 3.



**Figure 3.** The DQN-based offloading and compression decision process in Autonomous Vehicles [54] (Park et al., 2020).

Due to limitations in server capacity, internal policies within the MEC are required to influence the multimedia offloading decision. Deep Q-Learning has been found to be a tool capable of offering maximized reward for offload functions. Some of these advantages are credited to the operation of the layered structure DQN which is able to perform learning operations from small sections of agent data. Assessment of offloading and compression decision is performed in terms of state, action and reward. The state of offloading is expressed by the vehicle's capacity and standBy Q capacity while that of the compressing decision is represented by standBy Q and MEC capacity. Offloading delays

and energy consumption are the mark of how rewarding the offloading decision was. The data quality and waiting delay are responsible for establishing the reward mechanism for the compression decision. The outcomes of the performance appraisal show that DQN, as in many other processes, quickens the offloading and compression in autonomous vehicles operating in highly dynamic environments.

In Ban et al., 2020 [55], the authors developed a 360-degree (virtual reality) video streaming service which employs deep reinforcement learning for prediction and allocation of streaming resources. Their scheme solved the problem involved in multi-user live VR video streaming in edge networks. To deliver consistent video quality across all users, the server requires higher bitrates to cope with data sizes related to delivery of VR videos due to its spherical nature. The system utilized the Mean Field Actor-Critic (MFAC) algorithm to enable the server to collaborate and distribute video segments on request to maximize the general quality of experience while reducing bandwidth utilization. The deployment of edge cache network enables multiple users to be served concurrently. The utilization of an edge-assisted framework helps to minimize congestion on the backhaul network. The client changes their title rates to improve both the quality of experience and the total bandwidth requirement by communicating over several edge servers. The authors used the Long Short-Term Memory (LSTM) network to forecast user's future bandwidth and viewing activities to adapt the dynamic network and playback settings.

The authors utilized the multi-agent deep reinforcement learning (MADRL) model to tackle the problem associated with high-dimensional distributive collaboration and to study the optimal rate allocation scheme. The objective of the virtual reality video streaming scheme concentrates on four aspects, namely, average quality, temporal viewing variance, playback delay and bandwidth consumption. The performance evaluation of MA360's with 48 users on different live video was executed over three experiment labels from video number 1 to 3 distinctly. From the evaluation, as the video number increased, the normalized quality of experience remained fixed for all methods and the download traffic increased, respectively. The MA360 scheme could be easily transferred to the present streaming systems with variable number and video numbers. Simulations carried out on data derived from actual events were used to establish comparison between MA360 and some state-of-the-art streaming methods such as Standard DASH algorithm (SDASH), Leverages LR (LRTile), ECache, Pytheas. The result showed that MA360 improved the total quality of experience and reduced bandwidth consumption. It also showed that MA360 exceeded the current state-of-the-art scheme performance in terms of different network circumstances.

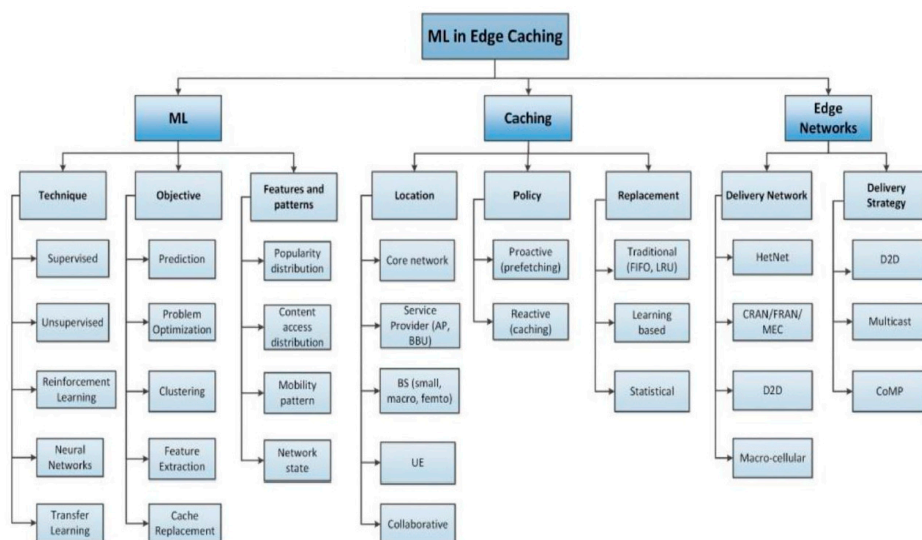
## 5. Multimedia Edge Caching and AI

Caching has become an integral part of computer networks globally. The need for the short-term storage of transient data with the exponentially growing traffic created by multimedia files such as video, images and other data types existing in virtual and local servers has made caching a necessity. In caching, subsets of data are maintained in a storage location within close proximity to the user to eliminate the repeated query operations channeled to the main source such as a cloud storage resource. The challenge that exists in this space involves deciding on what to cache, identifying where data are required and managing caching resources in a way that reflects the need and storage capacity to obtain trade-off benefits. In this section, we examine different research efforts targeted at the application of machine learning techniques within edge networks to identify and predict multimedia caching opportunities.

Edge networks have evolved to utilize in-network data caches which can be in the form of user equipment and in some cases base station installations, to manage the latency in backhaul created by distant central cloud storage resources (Wang et al., 2017) [56]. Content Distribution Networks (CDN) are notably the first instance where cache deployments are recognized and eventually become major contributors to 5G networks and possibly future deployments with mobile network operator-managed local infrastructure offering greater

caching capacity and higher backhaul performance marked by improved coverage (Wang et al., 2019 [57]; Yao et al., 2019 [58]). The application of machine learning in Mobile Edge Caching and other radio network instances promotes the predictive capacity within caching layers. This requires a capture of futuristic data demand leading to a reduced need for backhaul interaction for content access. In one study on the caching of videos within CDNs in MEC, Zhang et al., 2019 [59] applied a variant of recurrent neural networks which utilizes a deep Long Short-Term Memory network cell (LSTM-C) as a means of cache prediction and content update in a CDN, to optimize video caching in streaming. The methodology reveals improvement on previously existing caching algorithms such as the first in first out (FIFO), Least recently used (LRU) method among others.

Shuja et al., (2021) [60] presented a review of several intelligent data cache methods in edge networks. Considering the role of constantly evolving IoT and other multimedia devices which create a demand for low latency bandwidth supply capable of handling the loading of backhaul networks, the review comprehensively covered several machine learning variations and developed a taxonomy (shown in Figure 4) which accounted for applicable machine learning techniques, caching strategy and edge networks and how they work together to address the challenge of what, when and where to cache data. The benefits of this methodology were observed in the technological architecture of 5G technologies where leveraging millimeter-wave (mmWave), ultra-reliable low latency communication (URLLC), edge computing and data caching have greatly improved peak data rates for uplink and downlink processes. Further requirements beyond these tools are the need for increased efficiency in the management of limited network resources which may be achieved by network traffic prediction, the utilization of routing algorithms and the reduction in network congestion. The availability of large data sets and computing resources present in edge computing promote the opportunity for incorporation of various implementations of machine learning based on the unique efficiencies associated with them.



**Figure 4.** ML-Edge-Caching Taxonomy [60] (Shuja et al., 2021).

The increased performance capability of local devices and localized storage capacities provide an opportunity for caching without infrastructure in edge networks. Yao et al., (2019) [58] highlighted the extent of the impact caches have on backhaul links by addressing the caching process to identify challenges occurring within the four-phase process. The architecture of mobile edge caches is greatly influenced by the unique interactions shared by various caching options and predominant problems experienced within the requesting, exploration, delivery and update phases. The full array of in edge network cache options identified in the review included user equipment (UE), base stations with differing capacity



variations, baseband unit pools and Cloud Radio Area Networks and mobile network infrastructure, and established joint multi-tier caching infrastructure.

Said et al., 2018 [61] researched the application of the Clustering Coefficient based Genetic Algorithm (CC-GA) for community detection with device-to-device communication integration. The machine learning cluster capability provides proactive cache opportunities which outperform reactive caching in terms of captured overall user experience. The benefits of adopting the Edge network architecture to involve multi-layer caching have been shown to reduce backhaul load (Sutton, 2018) [62]. A problem with the performance of backhaul networks is the requirement for repeat downloads of redundant multimedia data which create requests that are repetitive in nature. This leads to a backhaul loaded with redundant content requests. These challenges have been met with several alternative optimizations ranging from the proactive time-based content distribution network setup to offer transit linkage during periods of predicted congestion (Muller et al., 2016) [63] and reactive content caching as shown in Figures 5 and 6.

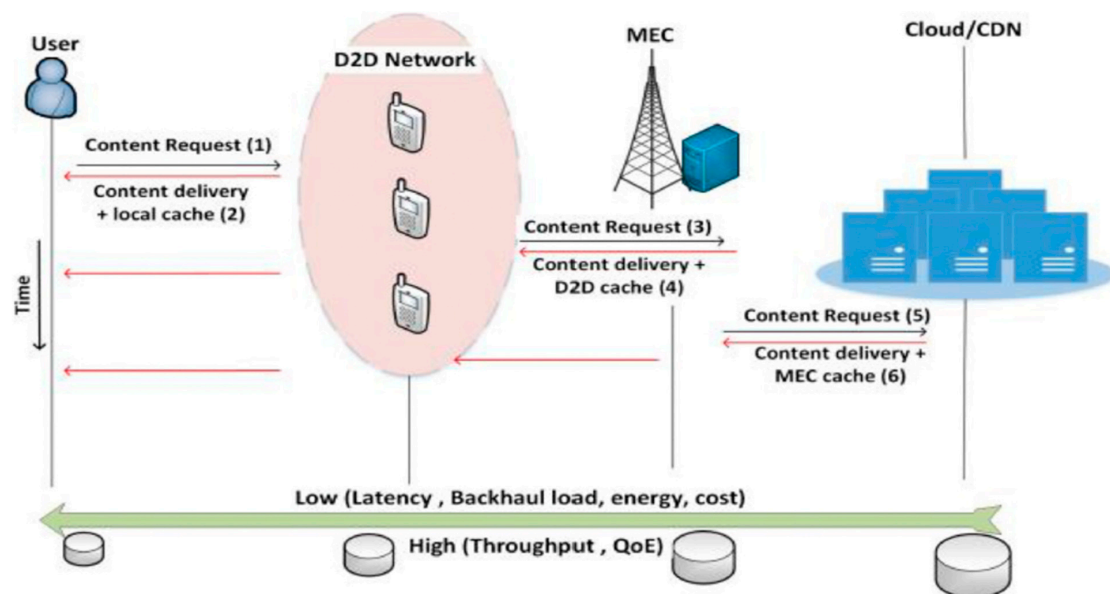


Figure 5. Reactive caching [60] (Shuja et al., 2021).

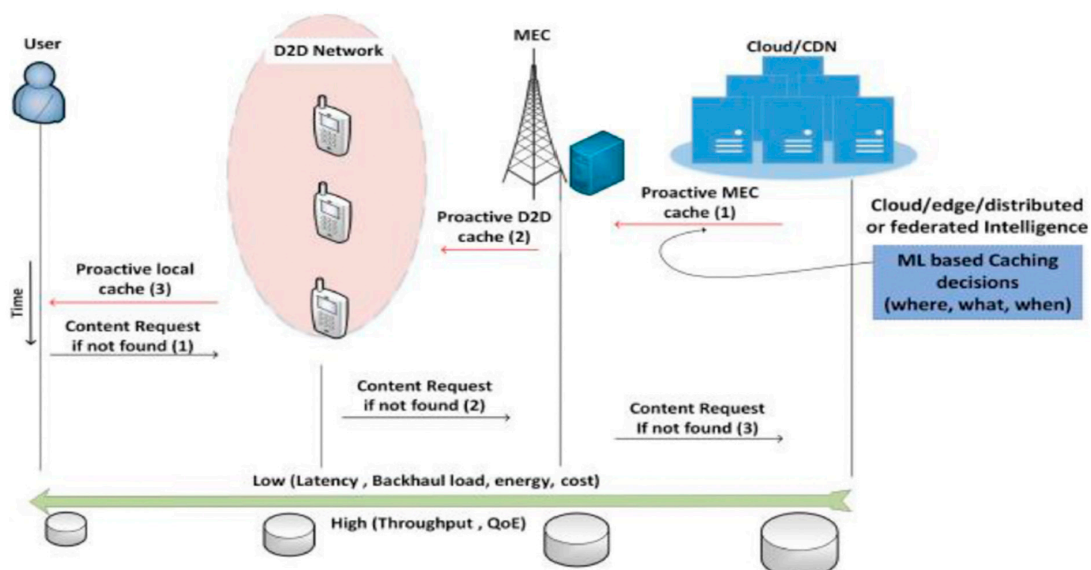


Figure 6. Proactive caching [60] (Shuja et al., 2021).



The anticipative and responsive cache options utilizing machine learning tools are largely affected by privacy policies, insight restrictions and complex user preference mapping. The identification of edge-specific trends can achieve a multi-process data-based user profiling in edge networks. Shuja et al., (2021) [60] established that the limits to machine learning in identifying user clusters depend largely on cache policy restrictions. Liu and Yang, 2019 [64] showed how deep reinforcement learning may be applied in proactive content caching on a deep-Q network. This outcome was achievable by applying learning derived from implemented recommendation policies expressed as two reinforcement learning problems run on a double deep-Q network. Wang et al., 2020 [65] also considered a Q-learning network to provide a model solution that offers flexible integrated multimedia caching between user equipment and network operator facilities in a heterogeneous nodal setting. In this method description, federated deep reinforcement learning is used to reactively enhance the Q-learning network by a multistage modelled system involving popularity prediction, device-to-device sharing within physical and social domains, and enhanced delay and transition models.

A popular theme in edge network content caches using machine learning is the application of reinforcement learning to create a multi-solution approach for caching requirements. Research on proactive content cache based on predicted popularity has been carried out by Doan et al., 2018 [66] and Thar et al., 2018 [67] with the former considering extracted raw video data mapped into G-clusters and analyzed by a predictor based on a convolutional neural network learning model to determine how much the content features deviate from a predefined ideal. Thar et al., 2018 [67] and Masood et al., 2021 [68] approached the challenge by utilizing deep learning to predict popularity scores. The former research applied class labels for content and assign them, while the latter applied a regression-based approach in its predictive functions. Based on the predictive machine learning model, content is then dispatched along the edge network to be cached at locations promoted by their popularity scores. Liu et al., 2020 [69] adopted a similar approach but went beyond the application of content popularity by applying a privacy preserving federated K-means led training for determining the appropriacy of content distribution along the edge network.

In more complex optimization situations such as observed in IoT communications, Xiang et al., 2019 [70] expressed a reactive methodology for caching within fog radio access networks (F-RANs) which utilized a deep reinforcement learning algorithm to prioritize user demands and allocate network resources. The methodology promoted core efficiency and transmission efficiency by slicing the network to cater to user categories as prioritized by the machine learning tool. In another work, Sun et al., 2018 [71] presented a reactive intelligent caching method combining Dynamic Adaptive Streaming over HTTP (DASH) made popular by YouTube with Deep Q-Learning for improved predictive efficiencies in video caching. The combination of both tools on a Mobile Edge Network can create an adaptive video caching service that responds reactively to changes along several variables identified by deep Q-learning. The authors identified the impact of buffer time losses within user equipment (UE) on overall perceived backhaul delays in video streaming within Mobile Edge Computing (MEC) caching schemes. On the network side, the loading effect on the backhaul is managed by fragmenting video files to bit sized data streams with information relevant for decoding, captured within a media presentation description (MPD). High density traffic along the backhaul network informs the intelligent caching along nodes referred to as agents within the network. A proactive application of deep learning was established by Masood et al., 2021 [68] which established a regression-based deep learning implementation on MEC storage devices which enabled video content prediction and mapping to multiple base stations across the edge network for caching purposes. Table 2 shows the evaluated caching research areas as a representative measure of achievable objectives hidden within various deployments of machine learning in edge network caching.

**Table 2.** Machine Learning Objectives in Edge Caching.

Machine Learning (ML) Methodology	ML Objective	Caching Objective	Caching Policy
Supervised (CNN) (Doan et al., 2018 [66])	Prediction of Content Popularity	Improvement of 5G Performance	Proactive
Supervised (DNN) (Thar et al., 2018 [67])	Prediction of Content Popularity and Content Distribution	Reduced backhaul latency	Proactive
Unsupervised (K-Means) (Liu et al., 2020 [69])	Clustering and privacy preservation	Reduced backhaul latency	Proactive
Supervised (Regression -based Deep Neural Network) (Masood et al., 2021 [68])	Prediction of Content Popularity	Reduced backhaul latency	Proactive
Supervised Deep Learning (Reinforcement Learning) (Xiang et al., 2019 [70])	Content Distribution	Location user prioritization and transmission efficiency	Reactive
Supervised (Attention-Weighted Federated Deep Reinforcement Learning) (Wang et al., 2020 [65])	Multimedia Popularity prediction and Content Distribution	Heterogenous caching	Reactive

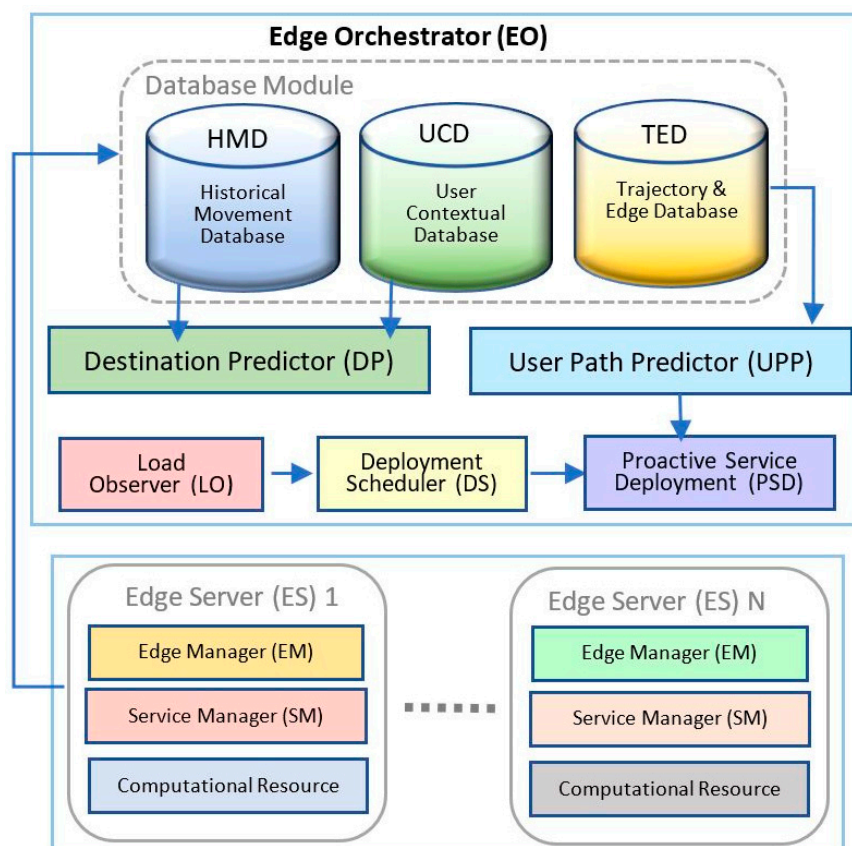
Mobile Edge Computing utilizing local user caches in some cases requires access to personal data to provide shared data caches for the promotion of content availability required for users within the boundaries of an edge network. One such case was investigated by Dai et al., 2020 [72] who captured the multimedia sharing challenges within vehicular edge computing (VEC). In this methodology, MEC base stations were adopted as verifiers of multimedia data obtained from vehicles which constitute the caching providers within the VEC network. The research highlighted how concerns around privacy protection shroud the willingness of users to have their data cache policy in VEC. To combat this challenge, blockchain driven permission systems were employed to ensure content is securely cached. This is achieved by users operating dynamic wallet addresses which leverage blockchain properties of anonymity, decentralization, and immutability. Content caching is optimized by the application of deep reinforcement learning (DRL) which manages caching operations despite changing wireless channels creating by vehicle mobility. Similar to other caching methods, these eases backhaul network traffic, utilizing vehicle to vehicle communication to reduce the demand associated with large multimedia transmissions. Their work investigated a cache requester and block verifier architecture utilized in a Manhattan city modelled grid, utilizing data from 4.5 million Uber pick-ups in New York City. The performance of deep reinforcement learning was evaluated using greedy content caching and random content caching which was plotted to generate a relationship between cumulative average reward for all requests and number of episodes initiated by caching requesters. The research showed the relationship between increased caching requesters and higher reward within a VEC. Another work by Li et al., (2019) [73] looked into cooperative edge caching as a means of eliminating redundant multimedia fetching protocols from base stations in MEC.

## 6. Multimedia Services for Edge AI

Multimedia service describes the interaction of voice, data, video and image in dual or multiple configurations taking place at the same time between the parties involved in some form of communication. Multimedia services can exist either as distributed or interactive services. Quality of Experience (QoE) is an important factor in multimedia services for users. Roy et al., (2020) [74] proposed mobile multimedia service driven by artificial intelligence in MEC, with the objective of achieving a high quality of experience.

The authors proposed an artificial intelligence-based method which utilizes meta-heuristic Binary Swarm Optimization (BPSO) to obtain a high performing solution. To manage and optimize nodes, an edge orchestrator (EO) managed by a Mobile network operator (MNO) makes use of statistical relationships derived from the nodal data and a mobility prediction model for planning multimedia service. The design assigns an edge server operating virtual machines responsible for managing user queries for each edge node creating multiple miniature data processing units. The EO has a controlling role of the edge server making use of the three database modules (movement data, contextual database and trajectory and edge database). The Path Oriented Proactive Placement (POPP) presents a twofold problem relating to the quality of user experience and the minimization of deployment cost in multimedia service delivery. The proposed POPP provides an intelligent interaction along the prediction path and optimizes Quality of Experience and cost reduction in real-time data processes.

The authors integrated the analysis of probabilistic relationships derived from historical movement data to predict and compensate for errors in the path prediction model. The implementation of the work was completed in cloudsim and a comparison was made with other existing works. The results indicated the performance of POPP exceeded those of previously existing work in QoE performance capturing superior satisfaction levels between 15% and 25% above deployments with similar objectives. The authors provided a computational model that expressed the POPP problem and developed a solution hinged on binary swarm optimization (BPSO) capable of managing the service placement requirements. Figure 7 shows the framework of the computational model utilized by the POPP system.



**Figure 7.** Computational framework of the POPP system [74] (Roy et al., 2020).

Wang et al., (2020) [65] developed research around an intelligent Deep Learning Reinforcement (DRL) edge-assisted crowdcast framework called DeepCast which examines

the total amount of viewing data for smart decisions to personalized Quality of Experience with minimized cost of system. Crowdcaster enables the viewers to watch and interact with the broadcaster and other viewers in a live video program. This interaction is completed in the same channel. The broadcasters use many platforms in crowdcaster services to stream their own content to the viewer; such platforms include Youtube, Gaming and Twitch.tv. Therefore, Crowdcaster faces challenges of poor Quality of Experience and high cost of services due to three major features in a crowdcaster service namely the crowdcaster platforms, content preferences and the rich interaction between the viewers and broadcaster.

The DeepCast which was proposed by the authors combined cloud, Content Distribution Network (CDN) and MEC for crowdcasting applications. Moreover, the Deepcast through the help of DRL recognizes the appropriate approach for allocation of viewers and transcoding on the edge server. The inherent process is data-driven and depends on the identification of complex trends from real-world datasets among components. To train the process, DRL is applied to trace-based experiments. Identified real world datasets applied in this process were obtained from inke.tv based in China having a viewership of up to 7.3 million users daily in 2016 and twitch.tv from the USA. Data fields are captured to represent users' datasets consists of the viewer and channel ID, network type, location, and viewing duration. Moreover, the collection of viewers' interaction information such as records of web application traffic, online exchanges and broadcaster's channel content from 300 well-known channels of Twitch.tv for two months was analyzed. In this framework, the responsibility of establishing a connection to the cloud server rests with the broadcaster creating a link that supports streaming of raw data. Streamed data are then encoded and compressed into chunks with multiple bitrates, which are conveyed to the content distribution network server. The DRL tool performs the function of allocating content with different bitrates to the relevant Edge servers based on QoE policy established from training. The results from the evaluation of the DeepCast system showed an effective improvement in the average personalized Quality of Experience than the cloud CDN method. The author cited a cost reduction of between 16.7 and 36% which was achievable by the implementation of the model. In conclusion, the utilization of the edge servers in DeepCast can satisfy viewer's personalized and heterogeneous QoE demands.

When there is a need for offloading of storage and computing resources to the network edge, the network is faced with problems such as latency and underutilized bandwidth. Guo et al., (2019) [75] proposed an approach utilizing Deep-Q-network based multimedia multi-service quality of service optimization for mobile edge computing systems. The authors investigated a multi-service situation in MEC systems. The MEC offers three multimedia services; streaming, buffered streaming and low latency enhanced mobile broadband applications (eMBB) for edge users. The packets scheduling method and quality of service model in mobile edge computing system were analyzed. Whenever mapping is required for converting a packet into a quantity of service flow, the scheduler is required to prioritize the matching of available resource with quality-of-service characteristics. The consideration of 5G quality of service model enables the packet from different multimedia applications to be mapped into different Quality of Service flows in accordance with the quality-of-service requirements. As a solution, a QoS maximization problem was formulated by which requirements for scheduling the limited radio resource can be defined and computed. The application of the 5G quality of service (QoS) model was used for satisfying various QoS conditions in several service cases. The processing of each quality of service was performed individually by allocating the same QoS flow to packets with similar requirements. A reinforcement-based deep-Q learning method was utilized to allocate dynamic radio resources. The Deep Reinforcement Learning framework performance was monitored using the properties of state space, action space, state performance and reward function. A simulation was performed, and the results indicated that the Deep-Q-Network-based algorithm performed better than the other resource allocation algorithms.

Huo et al., (2020) [76] proposed an energy efficient model for resource allocation in edge networks applying deep reinforcement learning. Their case study considered

multimedia broadband services in the mobile network and addressed the challenge of the inefficient allocation of resources including bandwidth and energy consumption. Energy consumption for the system takes the form of transmission energy and basic energy which are both required to support the network flow. A simulation of the proposed work was carried out on three base stations with a significant number of active users. Four variations of user structures involving in one case three users, and in others four, five and six users were considered. The obtained results verified the effectiveness of the DRL-based scheme and its usefulness in catering to mobile user requirements while outperforming competing methods in energy-efficient resource allocation.

Wu et al., (2021) [77] researched video service enhancement strategies that guarantee that video coding rates are fairly distributed over user devices. The research considered video coding rates under the constraints of statistical delay and limited edge caching capacity. For the content delivery to match the quality-of-service requirements, two methods were highlighted, the first was, content caching to ensure the content is as close to user as possible. The second was video delivery which achieves an optimized performance by a sequenced scheduling of users based on an optimization policy established to improve the network. Both systems performed well in differing scenarios as several studies have attempted to hybridize the methods to derive the combined benefits. The author proposed a combined human–artificial intelligence approach capable of improving caching hit rates by more accurately predicting video cache requirements within the MEC network. The artificial intelligence component is responsible for learning user interest, movie attributes and ratings in terms of low-order and high-order features. This capability is made possible by the joint functioning of the factorization machine (FM) model and multi-layer perceptron (MLP) model. This information concerning user preferences and behavior is adopted by a designed socially aware model that takes individual preferences and models them into groups depicting a demographic of users with similar interests.

The video delivery policy is founded on the user's interest prediction and edge caching decisions. The optimization problem which is posed by limited caching resources within the MEC, and video coding rates is modelled by first identifying the delay violation probability. This gives rise to an analytically derived statistical delay guarantee model with a dual bisection exploration scheme to guide service delivery. The solution to the modelled optimization problem yields video coding rates that outperform other user-based logic methodologies. Coupled with the predictive competence of the hybrid human-machine intelligence, video caching complements the adopted service delivery method to create a more reliable bandwidth allocation structure. To test the suitability of the proposed method, data were obtained from Movielens, a web-based video recommendation software which recommends movies to users based on previous interest. The results from observed service simulations showed that increases in video coding rates were met with corresponding changes in maximum delay tolerance and probability of delay violations exceeding QoS stipulations. The reduced constraint on delay violations made allowance for higher video coding rates as it showed that reduced constraint creates convergence in video coding rates while approaching a mean channel capacity.

## 7. Hardware and Devices for Multimedia on Edge Intelligence

Edge computing has great applications for multimedia technology. Graphics processing units (GPUs), high-end Field Programmable Gate Arrays (FPGAs) and Tensor processing units (TPU) are some of the multimedia edge AI computing devices/platforms [78]. This section provides a discussion on hardware and devices for multimedia on edge intelligence. The reader can refer to the survey papers in [79,80] for further works on GPU and FPGA-embedded intelligence systems. Edge-based hardware devices for the deployment of AI and machine learning can be classified into the following types: (1) Application-Specific Integrated Circuit (ASICs) Chips—ASICs for AI applications are designed specifically to execute machine/deep learning algorithms and have the advantages of being compact in size with low power consumption. Some examples of ASICs for AI are the ShiDianNao [81]



and Google TPU. (2) Graphics Processing Units (GPUs)—GPUs have the advantages of being able to perform massive parallel processing to increase the throughput and are able to achieve a higher computational performance for AI algorithms/modules compared to conventional microprocessor/CPU-based architectures.

Some examples of GPUs for AI are the Nvidia Jetson and Xavier architectures [82]; (3) Field-Programmable Gate Array (FPGA)—FPGAs have the advantages of being re-configurable to give flexibility to implement custom AI architectures with lower energy consumption and higher security. An example of an FPGA device which is commonly used for AI acceleration is the Xilinx ZYNQ7000; and (4) Neuromorphic chips—these brain-inspired chips have the advantages of accelerating neural network architectures with low energy consumption. An example of a neuromorphic device which is commonly used for AI acceleration is the Intel Loihi [83]. It should be noted that neuromorphic approaches may utilize algorithms (e.g., spiking neural networks) which are different from conventional AI approaches.

There are various ways or modes in which edge AI models can be deployed as discussed by the authors of [84]: (1) Edge-Based Mode—in this mode, the edge AI device receives and sends the data to the edge server to perform the inference/prediction processing and returns the results to the edge AI device. This mode has the advantage that the edge server contains the centralized inference model for ease of deployment but has the disadvantages of latency depending on the network bandwidth. (2) Device-Based mode—in this mode, the edge AI device retrieves the inference model from the edge server and performs the prediction/inference task locally. This mode has the advantage that the inference processing does not rely on the network bandwidth but has the disadvantage of having a higher computational and memory requirement on the edge device. (3) Edge-Device mode—in this mode, the inference model is partitioned into multiple parts depending on the current factors such as network bandwidth and server workload. The information processing task is then shared between the edge device and the edge server. The mode has the advantages of flexibility and dynamic resource management. (4) Edge-Cloud mode—this mode has similarities with the edge-device mode when the edge device is highly resource constrained.

The authors in [85] considered the industrial Internet of Things (IIoT) over artificial intelligence (AI) applications and presented a discussion on edge AI technology. The work proposed a shared active transfer learning (SATL) design in which the open difficulties of edge AI applications for IIoT frameworks can be solved through training and testing. The work began with a briefing on smart edge AI, which is a mix of AI and edge computing, with an emphasis on model training for IIoT applications. The suggested SATL design focused on the three edge AI concerns listed: (1) Customization; (2) Adaptability; and (3) Preserving privacy by the use of AI, TL, and FL, respectively. Adaptability customizes the AI scheme by adjusting the number of labeled samples based on the task requirements. TL improves responsiveness by allowing the scheme to smartly harmonize the new learning routine, and FL ensures privacy by using a shared training approach in which the devices do not exchange any information. SATL attains superior precision with a smaller number of connected edge nodes, and the precision maintains at the top ranks even when the number of training samples is significantly reduced, according to simulation data. When compared with alternative state-of-the-art techniques, the SATL model's training procedure took much less time.

### 7.1. GPU-Based Edge Hardware, Systems and Devices

Graphics Processing Units (GPUs) are high-speed graphic rendering processors with many parallel cores of about 100s to 1000s cores. They provide high-performance computing and, in comparison to CPUs, have a bigger size and a higher power consumption. GPUs are highly suited for AI tasks due to their large number of tiny cores, which allows for both neural network training and AI inference. Civerchia et al. [86] demonstrated the efficiency of 5G-based low latency remote control and image processing using AI and GPUs to drive

SuperDroid Robots in all posts. The captured images by the SuperDroid Robots are sent to the image identification scheme through the 5G network via the robot rover. The image processing application's output is provided to a remotely controlled app, which via the 5G network data plane relays the instruction to the robot rover. Image recognition installations in two different ways were investigated. One of the image processing applications runs on a mini-personal computer central processing unit, while the other runs on the Jetson Nano GPU. The rover's ability to complete the slalom and cross the finish line without hitting any cones is the qualitative measuring performance criteria. Image transfer across the entire virtualized 5G network, image processing with image processing software, control resolution and the activation of robot controls all depend on the dual mode of the control chain. The quantity estimates for all parts of the two service line delays, such as travel time and return from the rover to the NGC N6 interface connector using user space probes such as ping, showed that the proposed design is efficient. as shown in Figure 8.

### 7.2. FPGA-Based Edge Hardware, Systems and Devices

Field Programmable Gate Arrays (FPGAs) are made up of an arrangement of a matrix of programmable logic containing customizable logic blocks (CLBs) coupled via configurable interconnects. FPGAs can be reconfigured to satisfy specific application or feature demands. The hardware allows engineers with programming experience to reprogram the device whenever the need arises. When a large degree of flexibility is required, these are the best options. The System-on-Chip FPGA (Soc FPGA) is a popular FPGA implementation approach that combines programmable logic with processor cores (e.g., ARM, MIPS). The work in [87] presented an FPGA accelerator for a broadcasting classification model based on broadcast linear classifiers for continuous deep learning analysis (CLDA). Xilinx Vitis 2020.1 and C++ HLS were the building blocks of the project. They target the Xilinx ZCU102 kit at 200 MHz speed. The hardware is controlled by an ARM processor-based host program. The obtained CoRE50 dataset showed that the proposed optimization solution results in significant reductions in latency, resource and energy usage. In all CLDA variants, the FPGA architecture beats the Nvidia Jetson TX1 GPU, with reduced delays of four and five times per specification accordingly over the GPU for CLDA Plastic Cov. The design can perform class progressive lifetime learning for object categorization when paired with a freezing Convolutional Neural Network scheme.

### 7.3. ASIC-Based Edge Hardware, Systems and Devices

Application-Specific Integrated Circuits (ASICs) are specialized logic designs that use a custom circuit library and have a low power consumption, speed and a tiny footprint. ASICs are recommended for devices that will run in very high volumes because they are time-consuming to design and more expensive than other solutions [88]. Fuketa and Uchiyama [89] proposed a custom chip-shaped AI chip that supports power-saving computer tools and the development of AI computing systems that use parallelism to accelerate neural network processing. These chips are known as cloud AI chips and are used for both training and orientation using models of deep neural network (DNN) where processing capacity is very high. The paper described the architecture of edge AI cloud chips and offered the tools that developers can use to create them. The researchers focused on image recognition tasks, which are common in CPS applications such as autonomous driving and factory automation with the aim of reducing computing precision using 32-bit floating-point (FP32) precision to enhance energy economy. The work used 16-bit FP (FP16) precision for training, and so the GPUs support FP16. The DNN model is made up of multiple layers of neural networks stacked on top of each other. The weighted sum of the input activations is used to produce the output activations.

The application of deep learning for video analytics is disadvantaged with high computational overhead as noted by the authors of [90]. The authors addressed the challenge by proposing an approach termed as FastVA, a framework that integrates video analytics for deep learning with neural processing unit (NPU) and edge processing in mobile and im-

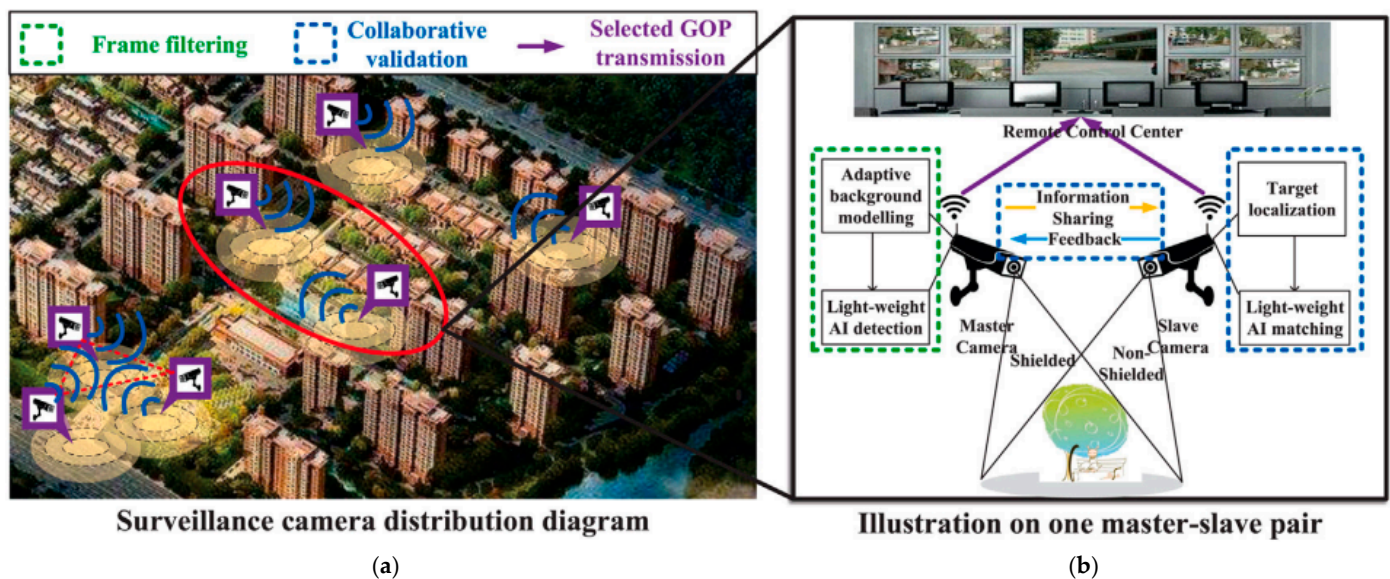
plemented FastVA on smartphones with extensive evaluations for its effectiveness. Based on the mobile application's accuracy and requirements, the project looked into several issues: (1) maximum accuracy, where the purpose is to achieve precision within a limited time; (2) maximum utility, where the purpose is to improve utility as an average dependant of precision and processing time; and (3) minimum energy, where the goal is to reduce energy usage under time and precision limitations. The authors discovered when to offload the work and when to employ NPU to overcome these challenges. Their method is based on the network condition, the NPU's unique properties and the optimization objective, which was presented as an integer programming problem with a heuristics-based scheme. To demonstrate the efficiency, the FastVA was deployed on smartphones for its evaluation.

#### 7.4. TPU-Based Edge Hardware, Systems and Devices

The Tensor Processing Unit (TPU) is a machine learning engine with a specific function. It is a processing IC created by Google to handle TensorFlow neural network computing. The integrated circuits are application-specific (ASICs) that are used to increase specific machine learning tasks by putting processing elements—small digital signal processors (DSPs) with inbuilt memory on a framework and allowing them to communicate and transport data between them. The study in [91] analyzed low-power computer topology built into ML-specific hardware in the context of Chinese handwriting recognition. The work used NVIDIA Jetson AGX Xavier (AGX), Intel Neural Compute Stick 2 (NCS2), and Google Edge TPU architectures have been tested for performance. The streaming latency of AlexNet and a bespoke version of GoogLeNet for optical character recognition were compared. Many architectures are not especially optimized for these models because they are custom-made and not commonly utilized. The AGX's massively parallel architecture allowed it to outperform the AlexNet model with more RAM. The TPU's neural network-optimized architecture allowed it to outperform the smaller-memory GoogLeNet model while avoiding high-end memory access penalties. Furthermore, because of its closely connected, ML-focused architecture, the NCS2 had the better average throughput compared with both design models, demonstrating its strong adaptation properties. TPU devices have been shown to work very well on the GoogLeNet model and in the intermediate state, according to the authors.

### 8. Use Case 1: Intelligent Multimedia Processing on Edge for Surveillance and Monitoring

The authors in [92] proposed a lightweight AI and IoT collaboration-based video pre-processing technique for wireless surveillance systems (Figure 8). The research employed a frame filtering module based on dynamic backdrop modeling with lightweight deep learning analysis. Their approach takes into account both static and dynamic surveillance circumstances. Unnecessary photographic groups (GOPs) in non-filtered videos are then filtered by selecting key frames that detect content on the edge cameras, which are acquired through a smart lightweight monitoring model to meet their limited processing capabilities, after dynamic backdrop modeling by the innovative operator of the proposed model. A slave camera detects and identifies obscure targets, providing feedback to the main camera for loading resolutions. The object acquisition model is developed using a step-by-step method, which combined channel pruning and convolutional acceleration. Ensuring the integration module between the surrounding cameras within the one-hop range compensates for the loss of accuracy caused by model shrinkage and environmental protection. The viability of this method was demonstrated by evaluations based on real-world videos. Between controlled trials and state-of-the-art approaches, it is shown that the proposed method may drastically reduce the transmission amount while maintaining a high degree of balance between system accuracy and delay. According to the evaluations, the method saved 64.4% of the bandwidth in a steady state and 61.1% in a flexible environment compared to green video transmission, making it a promising option.

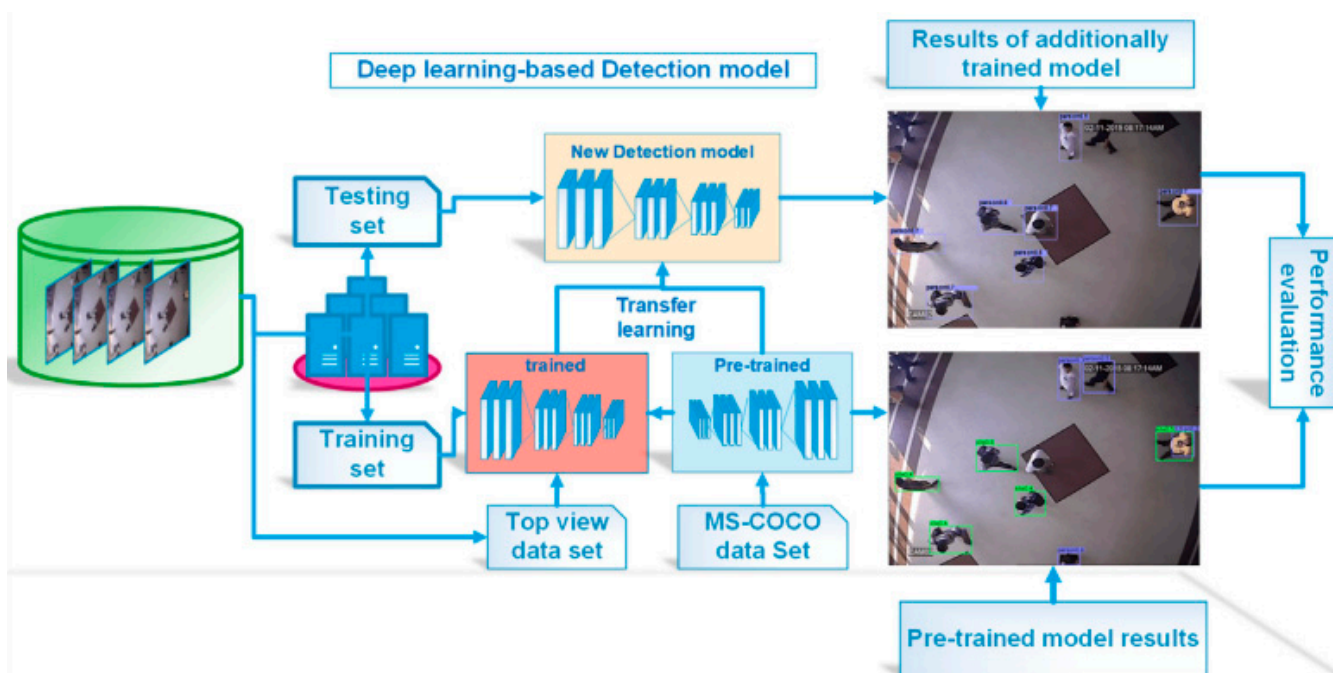


**Figure 8.** (a) Surveillance camera distribution view. (b) Illustration on a one master-slave pair [82].

Ahmed et al. [93] presented a real-world top view-based human recognition technique that uses a deep learning-based object recognition approach with only one stage, CenterNet, to recognize people (shown in Figure 9). The human being is recognized by the technique as a single point, sometimes referred to as the bounding box's center point. To acquire the center point, the design performs a key-point computation and reverts back all other properties of size, location and position information about the object of interest. The model was trained and tested on a top view data set in the work. Using the same information set, the recognition outcomes were compared to traditional recognition approaches. Physical edge, remote cloud and application layers are the three levels that make up the developed system. The high-quality photos are then sent to the layer for the cloud computing layer additional processing. The incoming data are analyzed in this layer before the data are sent to the video processing unit, which has minimal latency and increases the computing and calculations. The cloud computing layer was used to train the deep learning scheme. For human detection, the work used a CenterNet-based object recognition system. The work noted that because the visual elements of the human body vary significantly from top to bottom, more training is required to improve the recognition technique's effectiveness. The new and enhanced trained tier was connected up with the initial pre-trained algorithm and evaluated for the top view information set using transfer learning. With a pre-trained and trained model, the technique can achieve a recognition precision of 89% and 94%, respectively.

The work in [94] presented a Distributed Intelligent Video Surveillance (DIVS) system that used a deep learning (DL) scheme in an edge computing (EC) computing architecture. The work also created a multi-layer computer system for the DIVS system that included a distributed DL training technique. Their research projected a flexible data movement technique to resolve the mismatch in the throughput and processing capacity of the edge devices. To speed up the video analysis process, task-level parallel and design parallel training were applied by the authors. Parallel training, model synchronization and workload balance were all addressed in this paper. Two parallelization features were offered to boost the DIVS system's performance, and a key input value update method was forwarded to achieve global DL model compatibility. The use of compatible training methods at the work level was expected to speed up the video analysis process even further. In addition, the work recommended a model parameter review strategy to achieve global DL model synchronization in the distributed EC context. The test results showed that the EC design can provide faster and more uncomplicated processing power and that the formulated DIVS program was able to better manage the video surveillance and analytic functions.





**Figure 9.** Human recognition using a surveillance technique based on deep learning [93].

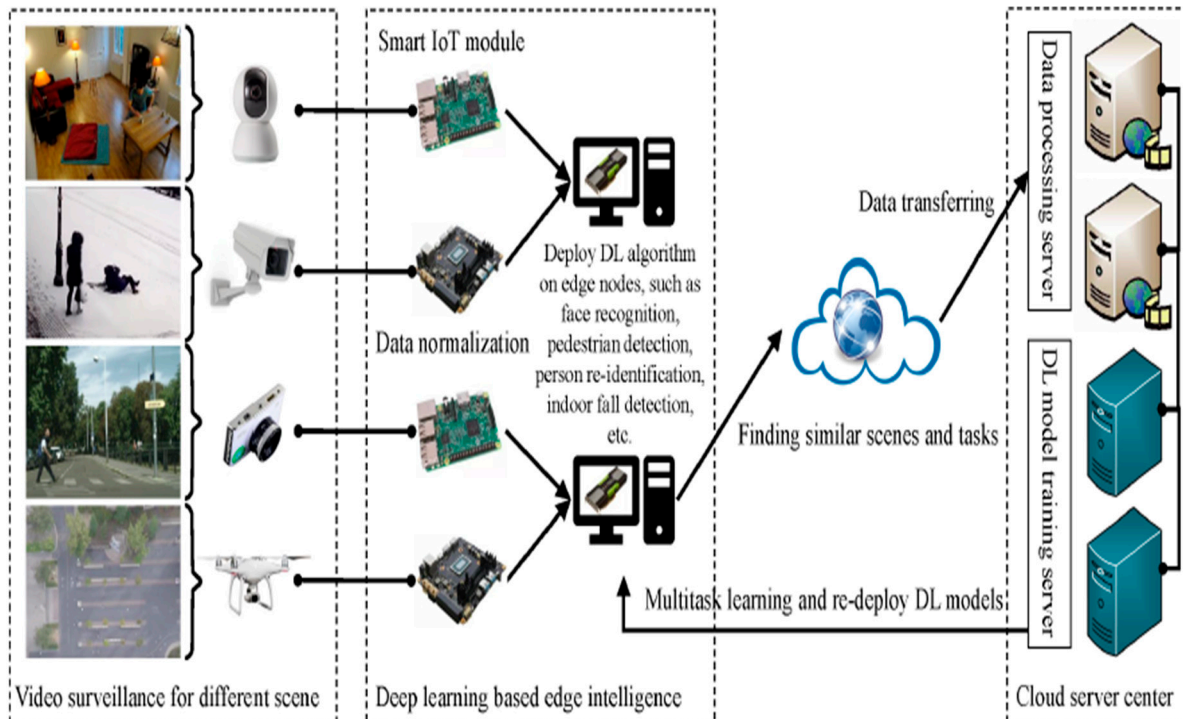
The authors in [95] proposed a smart video monitoring task allocation challenge that reduced the overall reaction time while achieving job performance expectations due to the inadequate computational power of the shared edge clouds with video surveillance. In addition, an adaptive deep neural networks (DNNs) model selection method was proposed, which compared the similarity between the feature and part of the included video and default training films to select each of the most effective DNN schemes. The authors showed that the smart video surveillance task orchestration challenge was NP-hard by minimizing the weighted reaction duration while achieving task efficiency demands. Their work offered an aware graph for a two-stage delay scanning strategy that achieved a good balance of network and computes delays. The experimental findings showed that their strategy was effective.

The paper in [96] presented a study on how to construct video surveillance systems using one of 5G's major technologies, mobile edge computing (MEC). The authors developed an image recognition scheme for the sensor of the camera and the MEC server to obtain a higher identification precision while keeping detection time to a minimum. The Q-learning strategy was used to train the system's actions by concurrently optimizing offloading decisions and image compression parameters to obtain a high recognition accuracy and a low recognition time. The design demonstrated how to weigh the benefits and drawbacks of utilizing extra decentralized computing power for a real-time video surveillance implementation with well-defined application-level demands, as well as how to manage the effect of lossy and rate-constrained wireless channels on the video surveillance implementation. The results of the experiments demonstrated the benefits of the recommended system design for providing network infrastructure, performance, and smart video surveillance.

The work in [97] proposed a customized Edge Intelligent Video Surveillance (EIVS) system (shown in Figure 10). It is a platform for robust edge computing that performs relevant computer vision tasks with multitasking deep learning. A smart IoT unit is used to standardize video data from many cameras in this work. The deep learning models were deployed at each EIVS node to perform computer vision tasks on normalized data. The work implemented the training depth classifier model jointly on a cloud server in a multitask approach. The research projected cooperated training of the deep learning models in a multifunctional paradigm on the cloud server for the related tasks in the same scenario, given that the training and deployment of deep learning models are normally



separated. The simulation findings on the known datasets showed that the system facilitates a smart surveillance of the activities constantly and in a robust manner and can boost efficiency using multitask learning.



**Figure 10.** The proposed edge intelligent video surveillance (EIVS) design architecture [97].

SurveilEdge, a collaborative cloud-edge system for real-time queries of large-scale surveillance video streams, was introduced by the authors of [98]. The work created a convolutional neural network (CNN) training method to shorten the training duration while maintaining a high accuracy, as well as a smart load balancer that balances the loads across multiple compute nodes and achieves the latency–accuracy tradeoff for real-time requests. The algorithms and system architecture in this paper were not confined to only a use case study in this work, and they can benefit most latency-sensitive deep learning applications. For future latency-sensitive applications, the proposed design offers a better compromise between delay, bandwidth cost and accuracy. The SurveilEdge was implemented on a prototype with numerous edge nodes and a public Cloud. SurveilEdge achieved up to 7 percent fewer bandwidth demands and 5.4% enhanced query response time than the cloud-only solution and can improve query precision by up to 43.9% and attain 15.8 speedups, accordingly, when compared to the edge other techniques.

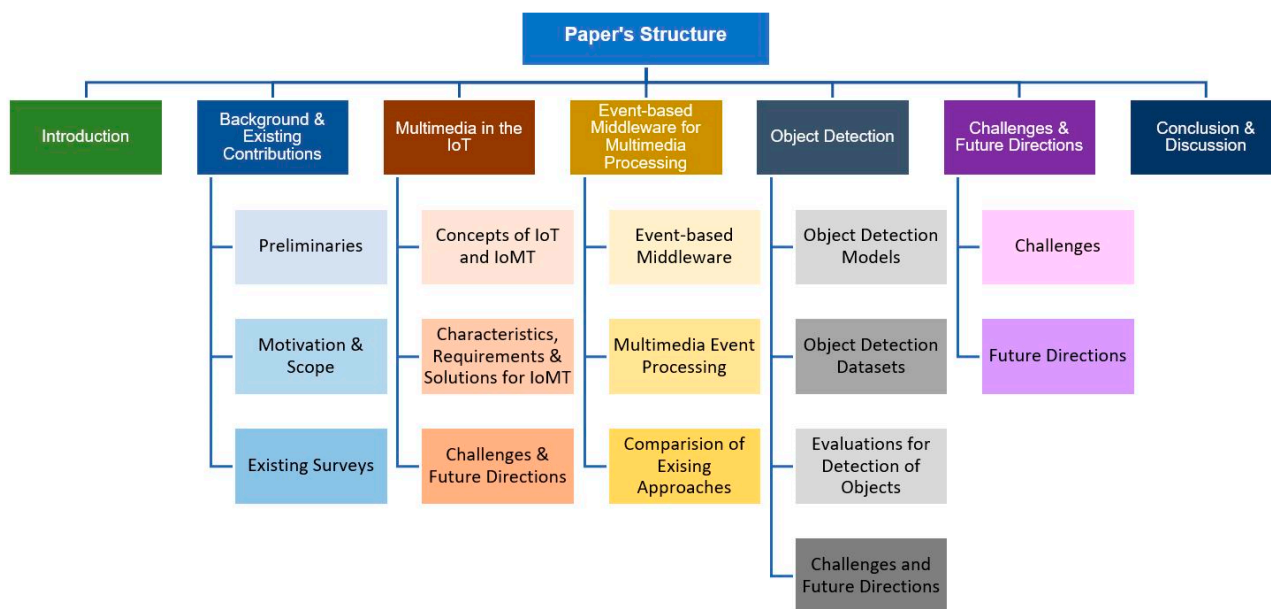
## 9. Use Case 2: Intelligent Multimedia Processing on Edge for Human Computer Interaction (HCI) and Health

The authors in [99] proposed a deep learning technique that may be efficiently implemented on portable gadgets to handle the difficulties in face detection and emotion recognition tasks. The work demonstrated a TensorFlow automation utilizing a Raspberry Pi microcontroller with an Intel Movidius Neural Compute Stick (NCS). The approach used in this work eliminates some of the setup processes that are required with TensorFlow, reducing the time spent on early elaboration notwithstanding the limited lag that exists when sending data to and from the Raspberry Pi controller. The findings provide a good starting point for further research into many optimization schemes for use in an embedded framework, including in the presence of expensive Deep Learning schemes, without the use of super expensive devices, such as GPU-accelerated hardware, which typically requires

more power resources. As a result, the NCS can accelerate the pipeline and deliver nearly real-time outputs (22 fps).

The authors in [100] proposed a methodology for identifying human activity on a real-time basis. The suggested activity recognition model includes a one-of-a-kind change detection module. They created an estimating model based on the differences between consecutive video frames rather than a binary classification such as changed/unchanged. The frame is categorized as altered based on the level of change, showing that the frame indicates a new activity. To complete such a task, a graphic accelerator optimization criterion was utilized, and thus the work produced a novel enhanced genetic method for recognizing the major differences between video frames. A deep learning method was used for recognizing the activity modified frame in the work. Running an effective activity identification model in a fog-assisted cloud makes it more efficient.

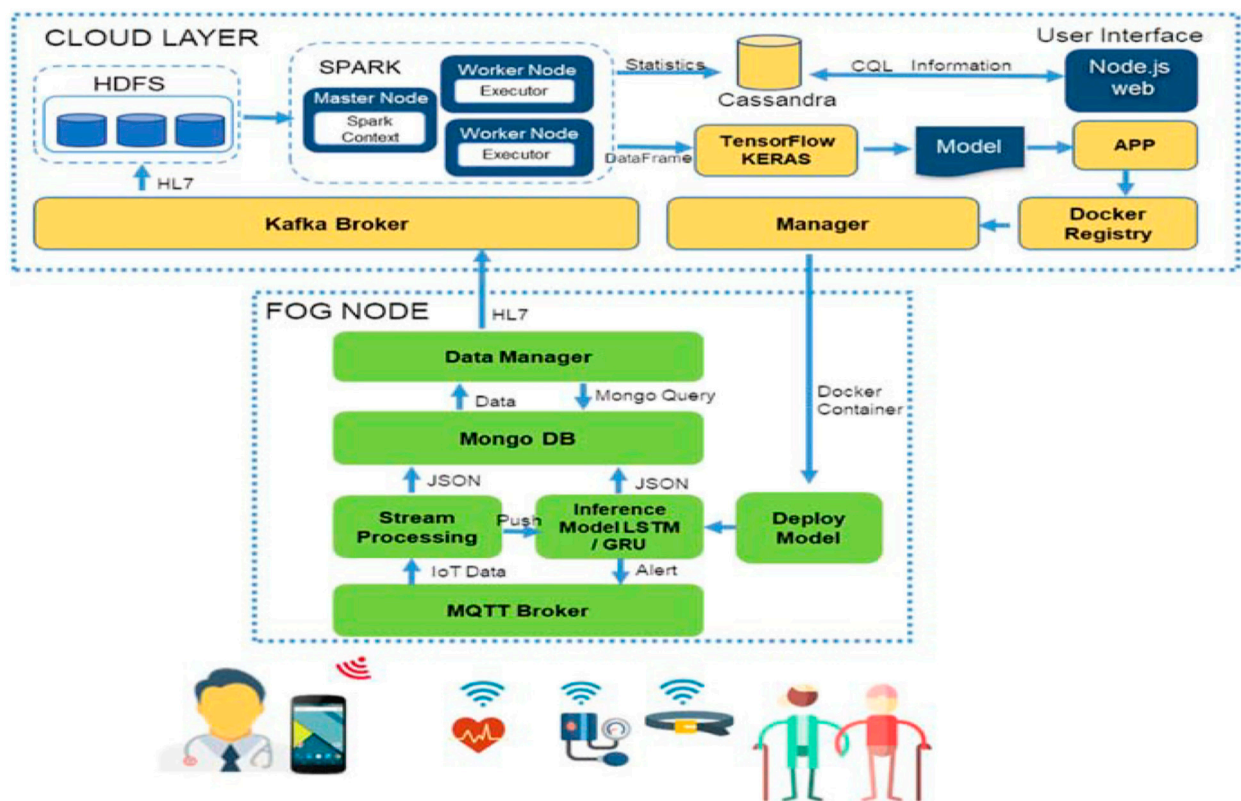
The work in [101] presented an overview of the problems and future research of generalized multimedia using the internet of things (IoMT) (shown in Figure 11), as well as a demonstration through an experiment that the existing processing multimedia events within smart cities models are very non-responsive to the hidden class, and the number of classes in the legacy rich datasets is insufficient to meet the demands of actual smart city technologies. The work also included an object identification case study to outline the requirements for processing multimedia events in smart cities and also used typical image recognition-based systems. The authors recommended an assessment of object detection schemes utilizing available datasets in order to validate the legacy challenges in object detection. At the end of each part, there was a brief discussion with highlighted gaps and a suggested strategy derived from past research studies, trials and evaluation. The work noted that there has not been much academic effort focused on the processing of multimedia events, although there is a significant amount of technical literature on the internet of everything and the research on IoMT is greatly increasing.



**Figure 11.** Survey structure of multimedia using the internet of things (IoMT) [101].

The authors in [102] proposed a novel, highly efficient intelligent system based on a fog-cloud computing architecture to detect falls in real-time utilizing deep learning techniques deployed on resource-constrained devices (fog nodes). Figure 12 shows the Healthcare detection workflow pipeline architecture. To acquire patient monitoring data, the researchers used a wearable tri-axial accelerometer (two tri-axial accelerometer sensors (ADXL345, MMA8451Q) and one gyroscope (ITG3200)). The wearable devices transmitted the captured accelerometer data to the fog nodes using the MQTT protocol. Their approach

used 2999 time-steps (equal to 15 s). The experiments were conducted using the Apache Spark platform and the Spark Dataframe library. The authors remarked that the system was tuned to avoid missing any real falls (i.e., in their system, having false positives was preferable to missing actual fall events). The work also developed a smart internet of things (IoT) Gateway architecture in the fog to facilitate the remote integration and control of deep learning (DL) models. The work proposed a smart-IoT-Gateway architecture in the fog to facilitate the remote configuration and control of DL models, and also used virtual machines to optimize resources, analyze the performance and inference time of two deep learning models. The results demonstrated that the efficacy of the fall system delivered a more quick and precise response than typical fall detector systems, as well as a higher efficiency of 98.75% accuracy, a shorter latency and an enhanced throughput.



**Figure 12.** Healthcare detection workflow pipeline architecture [102].

The objective of the work in [103] was to present a quick overview of the current state of the art in text-to-speech models and to assess the viability of applying these schemes on neural engines. This review is useful to anyone looking to deploy text-to-speech services within edge devices and serves as guidance in decision-making when it comes to selecting an appropriate model. The study reviewed the results of tests performed on a variety of vocoder models and assigned a voice quality rating to each feature-generating model. According to the authors, a proper ranking of these networks is impossible because they are all trained using various quantities of data for different timeframes on different machines to obtain their required outputs. The authors' judgment is based on listening to some voice samples produced for each pre-trained algorithm from their separate GitHub repositories, as well as reviews from the literature. The paper in [104] proposed the implementation of a full end-to-end edge computing system capable of reliably classifying hand motions acquired from thermography. A dataset from a thermography of 321 photos were developed, comprising of 321 thermal images for each sign language digit. The technique provided in this paper used real images from a poor-resolution thermal camera with 3232 picture quality. The data were loaded into a unique lightweight deep learning

system for hand gesture detection based on a limitation driven by deep residual learning. On the test data set, the developed system had a 99.52% accuracy, with the extra feature of being insensitive to backdrop light levels.

In [105], the authors presented a unique framework for implementing a proactive fallback mode of Voice over IP conversations by combining control-plane information accessible by 5G networks over multi-access edge computing (MEC). The system is based on lower layers providing channel state information (CSI) to the MEC via 5G NE, supporting the development of new context-aware solutions. The work envisions a machine learning (ML) provision that learns the user-context evolution relying on an access network of cell-specific radio signals, where the ML engine operates on a MEC host, and its prediction is utilized to adjust the network topology for a given application. The work evaluated the performance of a nonstationary-condition predictor (NSCP) depending on a NN in a propagation channel that is being monitored, showcasing its benefits in cases that the channel quality fluctuates abruptly due to non-stationarity propagation state, such as those that occur in the harmony with line-of-sight (LOS) spot and a non-LOS (NLOS) of the transitions (and vice versa) to be effective. The over-the-top VoIP network operators can use the ML engine's user-context prediction to energize fallback to a CS technique, which will give better customers quality of experience by eliminating voice over internet protocol call freezing.

In [106], the authors proposed an emotion communication system with increased encryption for protecting client private profile data in emotion-sensing applications. The authors described how to unlearn peoples' private data in voice renditions by utilizing an adaptive learning technique that can be applied at the edge. For decision-making, the representations with improved encryption can be transferred to a central server. Edge nodes are used in the proposed framework to encode speech information into a resilient and compact representation that can be broadcast through the network. By removing data regarding the author's gender and lingual recognition, the technique uses an adversarial learning framework to discover privacy-preserving characteristics. As a result, the suggested architecture is well-suited to real-world applications such as autonomous vehicles, e-health care and voice-based social aides. The developed model was tested on several spoken emotion datasets, demonstrating that it can encrypt users' personal birth information while improving the ability to identify emotions without compromising efficiency.

The research work in [107] proposed UbeHealth which is a pervasive healthcare initiative based on edge computing, deep learning, big data, high-performance computing (HPC) and IoT. The design has three major components and four layers that provide an improved network quality of service. Deep learning, big data and HPC were utilized to predict network traffic, which the Cloudlet and network layers then used to maximize data streams, data caching and routing tables. The traffic flows' application guidelines were categorized, allowing the network layer to better match application demand and detect malicious traffic and unusual data. A thorough literature review was conducted in the work to ascertain the design demands for the proposed system. The clustering technique was used for identifying the many types of data that come from the same application protocols. The framework was used to create the UbeHealth design proof-of-concept. The UbeHealth system was evaluated using three frequently used data sets. A publicly available dataset and codes for research in multimedia and edge AI can be found in [108].

The previous sections have discussed multimedia and edge AI architectures, and the techniques for increasing their performances. We now offer some final observations and a summary for future prospects, recommendations for future work and research directions in multimedia and edge AI/ML: (1) First, the computational complexity and latency requirements for edge AI devices need to be improved to support practical deployment of multimedia and edge AI devices. This is particularly important for real-time multimedia edge AI systems. The optimal balance between performing computations on the edge devices and/or the cloud needs to be researched further and quantified. (2) A second challenge and recommendation for future work is towards a reduction in the power con-



sumption and an increased energy efficiency of the devices. This can be achieved through the realization of edge AI devices using specialized hardware architectures and processors. (3) New trends in AI and ML also need to be researched further. A new trend for AI algorithms, architectures and techniques is towards explainable AI (XAI). One objective of XAI systems is to validate the behavior of an AI/ML system and to give guarantees that the system will perform as expected when deployed in a real-world environment [109]. The XAI is an important issue which has yet to be comprehensively resolved, as traditional and deep neural network architectures do not offer the benefits of transparency which have high importance in safety-critical applications (e.g., autonomous vehicles). Some works on XAI can be found in [110,111]. The authors in [110] discussed a XAI framework and a pipeline workflow which consists of three stages: (1) Model Understanding; (2) Model Diagnosis; and (3) Model Refinement. The authors in [111] give some recent work to explain the decision-making process and enhance the confidence of DNN-based solutions. In their work, the authors investigated DNN reactions towards predefined constraints and conditions for time series data. The authors postulated that their proposed approach could lead towards the development of dynamic and distributed AI-based systems for edge devices that have the benefits of XAI and trustworthiness. The expectation is that the XAI trend will carry on and play an important role in future architectures for multimedia edge AI and systems. Table Abbreviations and Acronyms gives a summary of the abbreviations and acronyms in the paper.

## 10. Conclusions

The large and increasing volumes of multimedia data which are being generated requires new techniques, approaches and hardware devices to be designed and deployed to be able to effectively realize the key targets for urban and smart city ecosystems. This has resulted in several challenges to existing AI and multimedia systems. Promising solutions to address these challenges are towards the development of the edge paradigm and models for multimedia AI. This paper has given a comprehensive survey of the research area of edge information processing techniques for multimedia and AI information processing. The review has covered several technological perspectives including multimedia analytics on edge empowered by AI, multimedia streaming on intelligent edge, multimedia edge caching and AI, multimedia services for edge AI, and hardware and devices for multimedia on edge intelligence. The paper has also discussed representative use cases for utilizing and deploying the edge paradigm for multimedia AI applications. Some recommendations for future work in multimedia and edge AI/ML have also been given. A summary of these recommendations is as follows: (1) reducing the computational complexity and latency requirements for edge AI devices to support practical deployment of multimedia and edge AI devices; (2) increasing the energy efficiency and lowering the power consumption of the edge devices; (3) developing distributed and federated learning algorithms for preserving user privacy; and (4) developing XAI algorithms, architectures and techniques to validate the behavior of edge AI/ML systems and to give guarantees that the system will perform as expected when deployed in a real-world environment. The challenges for computational and energy-efficient multimedia edge AI devices for architectures and techniques remain to be comprehensively resolved. The trend of XAI with multimedia and edge devices gives several new opportunities for researchers and practitioners.

**Author Contributions:** Conceptualization, J.K.P.S. and K.L.-m.A.; writing—original draft preparation, J.K.P.S., K.L.-m.A., E.P. and A.M.; writing—review and editing, J.K.P.S. and K.L.-m.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.



## Abbreviations and Acronyms

AE	Auto encoder
AI	Artificial intelligence
ASIC	Application specific integrated circuit
BN	Batch normalization
CCGA	Clustering Coefficient based Genetic Algorithm
CDN	Content distribution network
CNN	Convolutional neural network
CS	Compressed sensing
DASH	Dynamic adaptive streaming over HTTP
DEAL	Deployment Environment Aware Learning
DFIG	Data Fusion Information Group
DFT	Discrete Fourier Transform
DIVS	Distributed Intelligent Video Surveillance
DNN	Deep neural network
DQN	Deep Q-network
DRL	Deep reinforcement learning
DTMN	Discrete time Markov chain
EIVS	Edge Intelligent Video Surveillance
FANET	Flying ad-hoc network
FIR	Finite impulse response
FL	Federated learning
FPGA	Field programmable gate array
F-RAN	Fog radio access network
GAN	Generative adversarial network
GPU	Graphic processing unit
HAS	HTTP adaptive streaming
HCI	Human computer interaction
HMD	Head mounted display
HPC	High performance computing
IIR	Infinite impulse response
IIoT	Industrial internet of things
IoAV	Internet of autonomous vehicles
IoT	Internet of Things
IoV	Internet of vehicles
LAN	Local area network
LSTM	Long short-term memory
MADRL	Multi-agent deep reinforcement learning
MDP	Markov decision process
MEC	Mobile edge computing
MEVAO	Multi-user edge assisted video analytics
ML	Machine learning
MLP	Multilayer perceptron
NCS	Neural compute stick
NPU	Neural processing unit
PCA	Principal component analysis
POPP	Path Oriented Proactive Placement
QoE	Quality of experience
QoS	Quality of service
RAN	Radio access network
RBC	Radio bearer controls
RBM	Restricted Boltzmann machine
RELU	Rectified linear unit
RNN	Recurrent neural network
SDMN	Software defined mobile network
SDN	Software defined network

SUMO	Simulation of urban mobility
TL	Transfer learning
TPU	Tensor processing unit
VEC	Vehicular edge computing
VFC	Virtual function crossing
VFO	Virtual function orchestrator
VTS	Vertical tier scalability
XAI	Explainable artificial intelligence

## References

- Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall: Hoboken, NJ, USA, 2002.
- Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [[CrossRef](#)] [[PubMed](#)]
- Grosky, W.I. Multimedia information systems. *IEEE Multimed.* **1994**, *1*, 12–24. [[CrossRef](#)]
- Chew, L.W.; Chia, W.C.; Ang, L.M.; Seng, K.P. Low-memory video compression architecture using strip-based processing for implementation in wireless multimedia sensor networks. *Int. J. Sens. Netw.* **2012**, *11*, 33–47. [[CrossRef](#)]
- Silva, B.N.; Khan, M.; Han, K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustain. Cities Soc.* **2018**, *38*, 697–713. [[CrossRef](#)]
- Ang, L.M.; Seng, K.P.; Zungeru, A.M.; Ijamaru, G.K. Big sensor data systems for smart cities. *IEEE Internet Things J.* **2017**, *4*, 1259–1271. [[CrossRef](#)]
- Mehmood, Y.; Ahmad, F.; Yaqoob, I.; Adnane, A.; Imran, M.; Guizani, S. Internet-of-things-based smart cities: Recent advances and challenges. *IEEE Commun. Mag.* **2017**, *55*, 16–24. [[CrossRef](#)]
- Varghese, B.; Wang, N.; Barbhuiya, S.; Kilpatrick, P.; Nikolopoulos, D.S. Challenges and opportunities in edge computing. In Proceedings of the 2016 IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, USA, 18–20 November 2016; pp. 20–26.
- Ang, L.M.; Seng, K.P. Big sensor data applications in urban environments. *Big Data Res.* **2016**, *4*, 1–12. [[CrossRef](#)]
- Steiglitz, K. *Digital Signal Processing Primer*; Courier Dover Publications: Mineola, NY, USA, 2020.
- Han, Z.; Liu, Z.; Tang, J.; Gao, B.; Zhang, Y.; Qian, H.; Wu, H. Memristor-based signal processing for edge computing. *Tsinghua Sci. Technol.* **2021**, *27*, 455–471.
- Maleki, A.; Rashtchi, V.; Mazloun, J. Design and simulation of an infinite impulse response (IIR) filter with memristor. *Majl. J. Electr. Eng.* **2018**, *12*, 23–34.
- Hu, F.; Hao, Q. *Intelligent Sensor Networks: The Integration of Sensor Networks, Signal Processing and Machine Learning*; Taylor & Francis: Abingdon, UK, 2012.
- Bonomi, F.; Milito, R.A.; Zhu, J.; Addepalli, S. Fog computing and its role in the Internet of Things. In Proceedings of the 1st Edition MCC Workshop Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; pp. 13–16.
- Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surveys Tuts.* **2017**, *19*, 23222358. [[CrossRef](#)]
- Verbelen, T.; Simoens, P.; de Turck, F.; Dhoedt, B. Cloudlets: Bringing the cloud to the mobile user. In Proceedings of the ACM MCS, Lake District, UK, 9 June 2012; p. 2936.
- Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.
- Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. *Proc. Mach. Learn. Res.* **2013**, *28*, 1310–1318.
- Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM fully convolutional networks for time series classification. *IEEE Access* **2017**, *6*, 1662–1669. [[CrossRef](#)]
- Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Processing Mag.* **2018**, *35*, 53–65. [[CrossRef](#)]
- Larochelle, H.; Mandel, M.; Pascanu, R.; Bengio, Y. Learning algorithms for the classification restricted Boltzmann machine. *J. Mach. Learn. Res.* **2012**, *13*, 643–669.
- Zhang, C.; Liu, Y.; Fu, H. Ae2-nets: Autoencoder in autoencoder networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2577–2585.
- Bengio, Y.; Lecun, Y.; Hinton, G. Deep learning for AI. *Commun. ACM* **2021**, *64*, 58–65. [[CrossRef](#)]
- Kelleher, J.D. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2019.
- Collins, A.G. Learning structures through reinforcement. In *Goal-Directed Decision Making*; Academic Press: Cambridge, MA, USA, 2018; pp. 105–123.
- Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Mag.* **2017**, *34*, 26–38. [[CrossRef](#)]
- Yang, Q.; Zhang, Y.; Dai, W.; Pan, S.J. *Transfer Learning*; Cambridge University Press: Cambridge, UK, 2020.

28. Sharma, R.; Biookaghazadeh, S.; Li, B.; Zhao, M. Are existing knowledge transfer techniques effective for deep learning with edge devices? In Proceedings of the IEEE International Conference Edge Computing (EDGE), San Francisco, CA, USA, 2–7 July 2018; pp. 42–49.
29. Chen, Q.; Zheng, Z.; Hu, C.; Wang, D.; Liu, F. Data-driven task allocation for multi-task transfer learning on the edge. In Proceedings of the IEEE 39th International Conference on Distributed Computer Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019.
30. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Mag.* **2020**, *37*, 50–60. [\[CrossRef\]](#)
31. Qu, C.; Calyam, P.; Yu, J.; Vandanapu, A.; Opeoluwa, O.; Gao, K.; Wang, S.; Chastain, R.; Palaniappan, K. DroneCOCONet: Learning-based edge computation offloading and control networking for drone video analytics. *Future Gener. Comput. Syst.* **2021**, *125*, 247–262. [\[CrossRef\]](#)
32. Ilhan, H.E.; Ozer, S.; Kurt, G.K.; Cirpan, H.A. Offloading deep learning empowered image segmentation from UAV to edge server. In Proceedings of the 2021 44th International Conference on Telecommunications and Signal Processing (TSP), Online, 26–28 July 2021; pp. 296–300.
33. Monburinon, N.; Zabir, S.M.S.; Vechprasit, N.; Utsumi, S.; Shiratori, N. A novel hierarchical edge computing solution based on deep learning for distributed image recognition in IoT systems. In Proceedings of the 2019 4th International Conference on Information Technology (InCIT), Bangkok, Thailand, 24–25 October 2019; pp. 294–299.
34. Munir, A.; Blasch, E.; Kwon, J.; Kong, J.; Aved, A. Artificial intelligence and data fusion at the edge. *IEEE Aerosp. Electron. Syst. Mag.* **2021**, *36*, 62–78. [\[CrossRef\]](#)
35. Kim, J.; Kim, N.; Won, C.S. Deep edge computing for videos. *IEEE Access* **2021**, *9*, 123348–123357. [\[CrossRef\]](#)
36. Jainuddin, A.A.A.; Hou, Y.C.; Baharuddin, M.Z.; Yusoff, S. Performance analysis of deep neural networks for object classification with edge TPU. In Proceedings of the 2020 8th International Conference on Information Technology and Multimedia (ICIMU), Selangor, Malaysia, 24–26 August 2020; pp. 323–328.
37. Chaitra, S.; Ghana, S.; Singh, S.; Poddar, P. Deep learning model for image-based plant diseases detection on edge devices. In Proceedings of the 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; pp. 1–5.
38. Tan, T.; Cao, G. Deep learning video analytics on edge computing devices. In Proceedings of the 2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Online, 6–9 July 2021; pp. 1–9.
39. Wu, W.; Gao, Y.; Zhou, T.; Jia, Y.; Zhang, H.; Wei, T.; Sun, Y. Deep reinforcement learning-based video quality selection and radio bearer control for mobile edge computing supported short video applications. *IEEE Access* **2019**, *7*, 181740–181749. [\[CrossRef\]](#)
40. Chen, Y.; Zhang, S.; Xiao, M.; Qian, Z.; Wu, J.; Lu, S. Multi-user edge-assisted video analytics task offloading game based on deep reinforcement learning. In Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, China, 2–4 December 2020; pp. 266–273.
41. Ran, X.; Chen, H.; Zhu, X.; Liu, Z.; Chen, J. Deepdecision: A mobile deep learning framework for edge video analytics. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 15–19 April 2018; pp. 1421–1429.
42. Peng, D.; Yuying, X.; Yun, S.; Huibin, D. Research on the application of 5G cloud-network-edge-device convergence and intelligent video technology in smart grid. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin, China, 28 June–2 July 2021; pp. 1286–1290.
43. Ali, M.; Anjum, A.; Yaseen, M.U.; Zamani, A.R.; Balouek-Thomert, D.; Rana, O.; Parashar, M. Edge enhanced deep learning system for large-scale video stream analytics. In Proceedings of the 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC), Washington, DC, USA, 1–3 May 2018; pp. 1–10.
44. Tsakanikas, V.; Dagiuklas, T. Enabling real-time AI edge video analytics. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Online, 14–23 June 2021; pp. 1–6.
45. Zhou, Y.; Xu, X.; Shen, F.; Zhu, X.; Shen, H.T. Flow-edge guided unsupervised video object segmentation. *IEEE Trans. Circuits Syst. Video Technol.* **2021**. [\[CrossRef\]](#)
46. Cheng, D.; Meng, G.; Xiang, S.; Pan, C. FusionNet: Edge aware deep convolutional networks for semantic segmentation of remote sensing harbor images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 5769–5783. [\[CrossRef\]](#)
47. Jiang, X.; Yu, F.R.; Song, T.; Leung, V.C. Intelligent resource allocation for video analytics in blockchain-enabled internet of autonomous vehicles with edge computing. *IEEE Internet Things J.* **2020**. [\[CrossRef\]](#)
48. Kristiani, E.; Yang, C.; Huang, C. iSEC: An optimized deep learning model for image classification on edge computing. *IEEE Access* **2020**, *8*, 27267–27276. [\[CrossRef\]](#)
49. Chang, Z.; Zhou, X.; Wang, Z.; Li, H.; Zhang, X. Edge-assisted adaptive video streaming with deep learning in mobile edge networks. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakech, Morocco, 15–19 April 2019; pp. 1–6.
50. Zhou, P.; Xie, Y.; Niu, B.; Pu, L.; Xu, Z.; Jiang, H.; Huang, H. QoE-aware 3D video streaming via deep reinforcement learning in software defined networking enabled mobile edge computing. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 419–433. [\[CrossRef\]](#)
51. Ali, J.; Roh, B.H. Management of software-defined networking powered by artificial intelligence. In *Computer-Mediated Communication*; IntechOpen: Vienna, Austria, 2021.

52. Luo, J.; Yu, F.R.; Chen, Q.; Tang, L. Adaptive video streaming with edge caching and video transcoding over software-defined mobile networks: A deep reinforcement learning approach. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 1577–1592. [\[CrossRef\]](#)
53. Dai, P.; Song, F.; Liu, K.; Dai, Y.; Zhou, P.; Guo, S. Edge intelligence for adaptive multimedia streaming in heterogeneous internet of vehicles. *IEEE Trans. Mob. Comput.* **2021**. [\[CrossRef\]](#)
54. Park, S.; Kang, Y.; Tian, Y.; Kim, J. Fast and reliable offloading via deep reinforcement learning for mobile edge video computing. In Proceedings of the 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 7–10 January 2020; pp. 10–12.
55. Ban, Y.; Zhang, Y.; Zhang, H.; Zhang, X.; Guo, Z. MA360: Multi-agent deep reinforcement learning based live 360-degree video streaming on edge. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
56. Wang, W.; Lan, R.; Gu, J.; Huang, A.; Shan, H.; Zhang, Z. Edge caching at base stations with device-to-device offloading. *IEEE Access* **2017**, *5*, 6399–6410. [\[CrossRef\]](#)
57. Wang, R.; Li, R.; Wang, P.; Liu, E. Analysis and optimization of caching in fog radio access networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8279–8283. [\[CrossRef\]](#)
58. Yao, J.; Han, T.; Ansari, N. On mobile edge caching. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2525–2553. [\[CrossRef\]](#)
59. Zhang, C.; Pang, H.; Liu, J.; Tang, S.; Zhang, R.; Wang, D.; Sun, L. Toward edge-assisted video content intelligent caching with long short-term memory learning. *IEEE Access* **2019**, *7*, 152832–152846. [\[CrossRef\]](#)
60. Shuja, J.; Bilal, K.; Alasmay, W.; Sinky, H.; Alanazi, E. Applying machine learning techniques for caching in next-generation edge networks: A comprehensive survey. *J. Netw. Comput. Appl.* **2021**, *181*, 103005. [\[CrossRef\]](#)
61. Said, A.; Shah, S.W.H.; Farooq, H.; Mian, A.N.; Imran, A.; Crowcroft, J. Proactive caching at the edge leveraging influential user detection in cellular D2D networks. *Future Internet* **2018**, *10*, 93. [\[CrossRef\]](#)
62. Sutton, A. *5g Network Architecture, Design and Optimization*; IET 5G Conference: London, UK, 2017.
63. Müller, S.; Atan, O.; van der Schaar, M.; Klein, A. Context-aware proactive content caching with service differentiation in wireless networks. *IEEE Trans. Wirel. Commun.* **2016**, *16*, 1024–1036. [\[CrossRef\]](#)
64. Liu, D.; Yang, C. A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users. *IEEE Access* **2019**, *7*, 83120–83136. [\[CrossRef\]](#)
65. Wang, F.; Zhang, C.; Wang, F.; Liu, J.; Zhu, Y.; Pang, H.; Sun, L. Deepcast: Towards personalized qoe for edge-assisted crowdcast with deep reinforcement learning. *IEEE/ACM Trans. Netw.* **2020**, *28*, 1255–1268. [\[CrossRef\]](#)
66. Doan, K.N.; van Nguyen, T.; Quek, T.Q.; Shin, H. Content-aware proactive caching for backhaul offloading in cellular network. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 3128–3140. [\[CrossRef\]](#)
67. Thar, K.; Tran, N.H.; Oo, T.Z.; Hong, C.S. DeepMEC: Mobile edge caching using deep learning. *IEEE Access* **2018**, *6*, 78260–78275. [\[CrossRef\]](#)
68. Masood, A.; Nguyen, T.; Cho, S. Deep regression model for videos popularity prediction in mobile edge caching networks. In Proceedings of the 2021 International Conference on Information Networking (ICOIN), Jeju Island, Korea, 13–16 January 2021; pp. 291–294.
69. Liu, Y.; Ma, Z.; Yan, Z.; Wang, Z.; Liu, X.; Ma, J. Privacy-preserving federated k-means for proactive caching in next generation cellular networks. *Inf. Sci.* **2020**, *521*, 14–31. [\[CrossRef\]](#)
70. Xiang, H.; Yan, S.; Peng, M. A deep reinforcement learning based content caching and mode selection for slice instances in fog radio access networks. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
71. Sun, C.; Zhou, J.; Zhou, X.; Zhang, X.; Wang, W. Deep learning enabled dynamic reactive video caching in mobile edge networks. In Proceedings of the 2018 IEEE International Conference on Communication Systems (ICCS), Chengdu, China, 19–21 December 2018; pp. 280–285.
72. Dai, Y.; Xu, D.; Zhang, K.; Maharjan, S.; Zhang, Y. Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4312–4324. [\[CrossRef\]](#)
73. Li, D.; Han, Y.; Wang, C.; Shi, G.; Wang, X.; Li, X.; Leung, V.C. Deep reinforcement learning for cooperative edge caching in future mobile networks. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakech, Morocco, 15–19 April 2019; pp. 1–6.
74. Roy, P.; Sarker, S.; Razzaque, M.A.; Hassan, M.M.; AlQahtani, S.A.; Aloï, G.; Fortino, G. AI-enabled mobile multimedia service instance placement scheme in mobile edge computing. *Comput. Netw.* **2020**, *182*, 107573. [\[CrossRef\]](#)
75. Guo, B.; Zhang, X.; Wang, Y.; Yang, H. Deep-Q-network-based multimedia multi-service QoS optimization for mobile edge computing systems. *IEEE Access* **2019**, *7*, 160961–160972. [\[CrossRef\]](#)
76. Huo, Y.; Song, C.; Ji, X.; Yang, M.; Yu, P.; Tao, M.; Shi, L. DRL driven energy-efficient resource allocation for multimedia broadband services in mobile edge network. In Proceedings of the 2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Paris, France, 27–29 October 2020; pp. 1–6.
77. Wu, D.; Bao, R.; Li, Z.; Wang, H.; Zhang, H.; Wang, R. Edge-cloud collaboration enabled video service enhancement: A hybrid human-artificial intelligence scheme. *IEEE Trans. Multimed.* **2021**. [\[CrossRef\]](#)

78. Wei, K.; Honda, K.; Amano, H. An implementation methodology for Neural Network on a Low-end FPGA Board. In Proceedings of the 2020 Eighth International Symposium on Computing and Networking (CANDAR), Naha, Japan, 24–27 November 2020; pp. 228–234.
79. Ang, L.M.; Seng, K.P. GPU-Based Embedded Intelligence Architectures and Applications. *Electronics* **2021**, *10*, 952. [CrossRef]
80. Seng, K.P.; Lee, P.J.; Ang, L.M. Embedded intelligence on FPGA: Survey, applications and challenges. *Electronics* **2021**, *10*, 895. [CrossRef]
81. Du, Z.; Fasthuber, R.; Chen, T.; Jenne, P.; Li, L.; Luo, T. ShiDianNao: Shifting vision processing closer to the sensor. In Proceedings of the 42nd Annual International Symposium of the Computer Architecture, Portland, OR, USA, 13–17 June 2015; pp. 92–104.
82. Nvidia Corporation. Jetson TX2 Module. Available online: <https://developer.nvidia.com/embedded/buy/jetson-tx2> (accessed on 1 May 2019).
83. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Wang, H. Loihi: A neuromorphic manycore processor with onchip learning. *IEEE Micro* **2018**, *38*, 82–99. [CrossRef]
84. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [CrossRef]
85. Foukalas, F.; Tziouvaras, A. Edge Artificial Intelligence for Industrial Internet of Things Applications: An Industrial Edge Intelligence Solution. *IEEE Ind. Electron. Mag.* **2021**, *15*, 28–36. [CrossRef]
86. Civerchia, F.; Giannone, F.; Kondepu, K.; Castoldi, P.; Valcarengghi, L.; Bragagnini, A.; Gatti, F.; Napolitano, A.; Borromeo, J.C. Remote control of a robot rover combining 5g, ai, and gpu image processing at the edge. In Proceedings of the 2020 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 8–12 March 2020; pp. 1–3.
87. Piyasena, D.; Lam, S.-K.; Wu, M. Edge accelerator for lifelong deep learning using streaming linear discriminant analysis. In Proceedings of the 2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Orlando, FL, USA, 9–12 May 2021; p. 259.
88. Haiming, Y.; Miyaoka, R.S.; Lewellen, T.K. A high-speed and high-precision Winner-Select-Output (WSO) ASIC. *IEEE Trans. Nucl. Sci.* **1998**, *45*, 772–776. [CrossRef]
89. Fuketa, H.; Uchiyama, K. Edge Artificial Intelligence Chips for the Cyberphysical Systems Era. *Computer* **2021**, *54*, 84–88. [CrossRef]
90. Tan, T.; Cao, G. Deep Learning Video Analytics Through Edge Computing and Neural Processing Units on Mobile Devices. *IEEE Trans. Mob. Comput.* **2021**. [CrossRef]
91. Kljucaric, L.; Johnson, A.; George, A.D. Architectural analysis of deep learning on edge accelerators. In Proceedings of the 2020 IEEE High Performance Extreme Computing Conference (HPEC), Online, 21–25 September 2020; pp. 1–7.
92. Liu, Y.; Kong, L.; Chen, G.; Xu, F.; Wang, Z. Light-weight AI and IoT collaboration for surveillance video pre-processing. *J. Syst. Archit.* **2021**, *114*, 101934. [CrossRef]
93. Ahmed, I.; Ahmad, M.; Rodrigues, J.J.; Jeon, G. Edge computing-based person detection system for top view surveillance: Using Center Net with transfer learning. *Appl. Soft Comput.* **2021**, *107*, 107489. [CrossRef]
94. Chen, J.; Li, K.; Deng, Q.; Li, K.; Yu, P.S. Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing. *IEEE Trans. Ind. Inform.* **2019**. [CrossRef]
95. Wu, Q.; Zhang, H.; Du, P.; Li, Y.; Guo, J.; He, C. Enabling adaptive deep neural networks for video surveillance in distributed edge clouds. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; pp. 525–528.
96. Hu, H.; Shan, H.; Wang, C.; Sun, T.; Zhen, X.; Yang, K.; Quek, T.Q. Video Surveillance on Mobile Edge Networks—A Reinforcement-Learning-Based Approach. *IEEE Internet Things J.* **2020**, *7*, 4746–4760. [CrossRef]
97. Li, J.; Zheng, Z.; Li, Y.; Ma, R.; Xia, S.-T. Multitask deep learning for edge intelligence video surveillance system. In Proceedings of the 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, UK, 20–23 July 2020; pp. 579–584.
98. Wang, S.; Yang, S.; Zhao, C. SurveilEdge: Real-time video query based on collaborative cloud-edge deep learning. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Online, 6–9 July 2020; pp. 2519–2528.
99. Hossain, M.S.; Muhammad, G. Emotion Recognition Using Deep Learning Approach from Audio-Visual Emotional Big Data. *Inf. Fusion* **2018**, *49*, 69–78. [CrossRef]
100. Subramanian, R.R.; Vasudevan, V. A deep genetic algorithm for human activity recognition leveraging fog computing frameworks. *J. Vis. Commun. Image Represent.* **2021**, *77*, 103132. [CrossRef]
101. Aslam, A.; Curry, E. A Survey on Object Detection for the Internet of Multimedia Things (IoMT) using Deep Learning and Event-based Middleware: Approaches, Challenges, and Future Directions. *Image Vis. Comput.* **2021**, *106*, 104095. [CrossRef]
102. Sarabia-Jácome, D.; Usach, R.; Palau, C.E.; Esteve, M. Highly-efficient fog-based deep learning AAL fall detection system. *Internet Things* **2020**, *11*, 100185. [CrossRef]
103. Bigioi, D.; Corcoran, P. Challenges for edge-ai implementations of text-to-speech synthesis. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 10–12 January 2021; pp. 1–6.
104. Breland, D.S.; Skriubakken, S.B.; Dayal, A.; Jha, A.; Yalavarthy, P.K.; Cenkeramaddi, L.R. Deep Learning-Based Sign Language Digits Recognition from Thermal Images With Edge Computing System. *IEEE Sens. J.* **2021**, *21*, 10445–10453. [CrossRef]
105. Centenaro, M.; Tomasin, S.; Benvenuto, N.; Yang, S. Predictive Voice-Over-Internet Protocol Fallback Over Vehicular Channels: Employing Artificial Intelligence at the Edge of 5G Networks. *IEEE Veh. Technol. Mag.* **2020**, *15*, 72–78. [CrossRef]



- 
106. Ali, H.S.; ul Hassan, F.; Latif, S.; Manzoor, H.U.; Qadir, J. Privacy enhanced speech emotion communication using deep learning aided edge computing. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Online, 14–23 June 2021; pp. 1–5.
  107. Muhammed, T.; Mehmood, R.; Albeshri, A.; Katib, I. UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities. *IEEE Access* **2018**, *6*, 32258–32285. [[CrossRef](#)]
  108. Edge-Dataset. Available online: <https://github.com/apgalano/Edge-Dataset> (accessed on 31 October 2021).
  109. Samek, W.; Montavon, G.; Vedaldi, A.; Hansen, L.K.; Müller, K.R. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*; Springer Nature: Berlin/Heidelberg, Germany, 2019; Volume 11700.
  110. Spinner, T.; Schlegel, U.; Schäfer, H.; El-Assady, M. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 1064–1074. [[CrossRef](#)]
  111. Dassanayake, P.; Anjum, A.; Bashir, A.K.; Bacon, J.; Saleem, R.; Manning, W. A deep learning based explainable control system for reconfigurable networks of edge devices. *IEEE Trans. Netw. Sci. Eng.* **2021**, *9*, 7–19. [[CrossRef](#)]