

Article

Brainware Computing: Concepts, Scopes and Challenges

Eui-Nam Huh  and Md Imtiaz Hossain * 

Department of Computer Science and Engineering, Kyung Hee University, Global Campus,
Yongin-si 17104, Korea; johnhuh@khu.ac.kr

* Correspondence: hossain.imtiaz@khu.ac.kr; Tel.: +82-31-201-2454

Abstract: Over the decades, robotics technology has acquired sufficient advancement through the progression of 5G Internet, Artificial Intelligence (AI), Internet of Things (IoT), Cloud, and Edge Computing. Though nowadays, Cobot and Service Oriented Architecture (SOA) supported robots with edge computing paradigms have achieved remarkable performances in diverse applications, the existing SOA robotics technology fails to develop a multi-domain expert with high performing robots and demands improvement to Service-Oriented Brain, SOB (including AI model, driving service application and metadata) enabling robot for deploying brain and a new computing model with more scalability and flexibility. In this paper, instead of focusing on SOA and Robot as a Service (RaaS) model, we propose a novel computing architecture, addressed as Brainware Computing, for driving multiple domain-specific brains one-at-a-time in a single hardware robot according to the service, addressed as Brain as a Service (BaaS). In Brainware Computing, each robot can install and remove the virtual machine, which contains SOB and operating applications from the nearest edge cloud. Secondly, we provide an extensive explanation of the scope and possibilities of Brainware Computing. Finally, we demonstrate several challenges and opportunities and then concluded with future research directions in the field of Brainware Computing.

Keywords: brainware computing; service oriented architecture (SOA); service oriented brain (SOB); brain as a service (BaaS)



Citation: Huh, E.-N.; Hossain, M.I. Brainware Computing: Concepts and Challenges. *Appl. Sci.* **2021**, *11*, 5303. <https://doi.org/10.3390/app11115303>

Academic Editor: Juan Francisco De Paz Santana

Received: 23 March 2021
Accepted: 2 June 2021
Published: 7 June 2021

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



Copyright: © 2021 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

Having Service Oriented Architecture (SOA) [1] and Collaborative Robots (COBOT) [2], the automation industry has grown to be a vast equalizer in diverse applications such as healthcare [3], medicine [4,5], e-commerce [6], surveillance systems [7], smart city [8], smart home [9], UAV [10], manufacturing [11], and so on [12]. Recently, the proliferation of 5G Internet, Artificial Intelligence (AI), Internet of Things (IoT), Cloud, and Edge Computing advance this automation technology further matured over the years yielding the noble services and technologies, for instance, Software as a Service (SaaS) [13], Infrastructure as a Service (IaaS) [14], Platform as a Service (PaaS) [15], Robot as a Service (RaaS) [16], etc. Moreover, these emerging scalable and flexible infrastructures drive the existing automation industry to accelerate the performance significantly in terms of perfection and resource utilization in business and defense.

The robotic concept was first introduced by Czech playwright Karl Capek in 1921 to denote a fictional humanoid [17,18]. Later in 1942, Isaac Asimov first used the term robotics [19]. After that, robotics technology has gradually improved in terms of performance, scalability, and flexibility through the progression of software and hardware technologies along with Networking, the IoT, AI, Cloud, and Edge Computing. To solve the ergonomic and productivity issues in the automation industry, Michael A. Peshkin et al. proposed a more flexible and scalable robot architecture called Collaborative Robot (COBOT) in 2001 [2]. However, Service-Oriented Architecture (SOA), Cloud Computing, and IoT push the margin of the automation industry services and scopes up, offering Robots as a Service (RaaS) [16] and Infrastructure as a Service (IaaS) [14]. In the last decade,

along with these service infrastructures and vast headways of Artificial Intelligence (AI) more specifically, deep learning and robotics technologies have attained gigantic capability in terms of accuracy and perfection on diverse applications [3–12].

Modern AI technology has obtained tremendous performances, but are all about domain-specific, hence state-of-the-art, robotics technology likewise. Every specific service or application demands a distinct domain-specific brain, i.e., artificial intelligent model and driving application. In this proposal, we use the keyword “brain” to denote a container image that includes an AI-trained model (if necessary, then, capable of being updated through service-aware learning), driving service application, and metadata. Developing a high-performing multi-domain expert system is extremely challenging. For example, a robot that is designed and trained for caring for a child is not capable of performing rationally as a nurse and vice versa. Yara Rizk et al. [20] explained the difficulties and challenges of multi-domain adaptive systems to describe the heterogeneity of the existing robotic systems. As a single robot owns a limited capacity of heterogeneity characteristics, Yunfei Shi et al. [21] considered the problem of a heterogeneous team of robots in terms of modeling and sampling. They proposed a technique to facilitate multiple robots working together. Aforementioned articles denote that existing single robotic systems struggle to perfectly support heterogeneity in terms of domain adaptation and performing better in diverse domains [20,21]. Deploying multiple brains inside a single robotic system might be one of the possible solutions to build the multi-domain adaptive robot. However, this solution is not practically and economically feasible because of the limited storage capacity and computation capability of the robots such as household robots, self-driving cars, etc.

To deal with the aforementioned issue, by the grace of the potential from 5G internet and possibilities from the state-of-the-art AI, edge, and cloud computing advancement, we propose a computing architecture named Brainware Computing. Brainware Computing denotes encapsulating AI model and service application with corresponding metadata in cloud, sharing domain specific brains to the edges, deploying and enabling requested brain images into the the robot based on the requested service, and combining and composing of multiple brains without external programming required. The existing edge computing paradigm has been considered as the backbone computing paradigm for Brainware Computing. Current robot architecture and working principles can be divided into four concerns: (1) Communication, (2) Storage, (3) Sensors and Actuators, and (4) Software. The software includes the AI model and driving application, which perform learning from the environment and inference of the next state by interacting with the real world environment [22] through sensors and actuators. In the existing robotics architecture, these four operations are standalone and fixed together. Though Robot as a Service (RaaS) [16] architecture introduced a strategy to increase the flexibility, it demands external control by the developers with very limited variability. In the Brainware Computing platform, a big portion of software component (learning, model updating and brain) is decoupled from the robot and makes the robot equipped to establish and exclude the virtual machine based on the service-demands (see Figure 1). The virtual machine includes Service Oriented AI model, corresponding operating applications and metadata. These virtual machines are stored in the form of the container image in the Intelligent Hierarchical Brain Store at the nearest edge cloud. Based on the service-demands, edge and core cloud can install and remove the service-specific brain image into and from the robots. Thus, a single robot can have a magnificent performance in any concerned domain by switching the intelligent brain inside it, which is being supervised by the edge cloud.

We elaborately discuss the need for Brainware Computing in Section 2. In Section 3, we show the concepts and introduction of the Brainware Computing with the case studies. Section 4 presents the scopes and possibilities. Challenges and opportunities of the proposed Brainware Computing are demonstrated in Section 5. Lastly, the conclusion along with the research direction are described in Section 6.

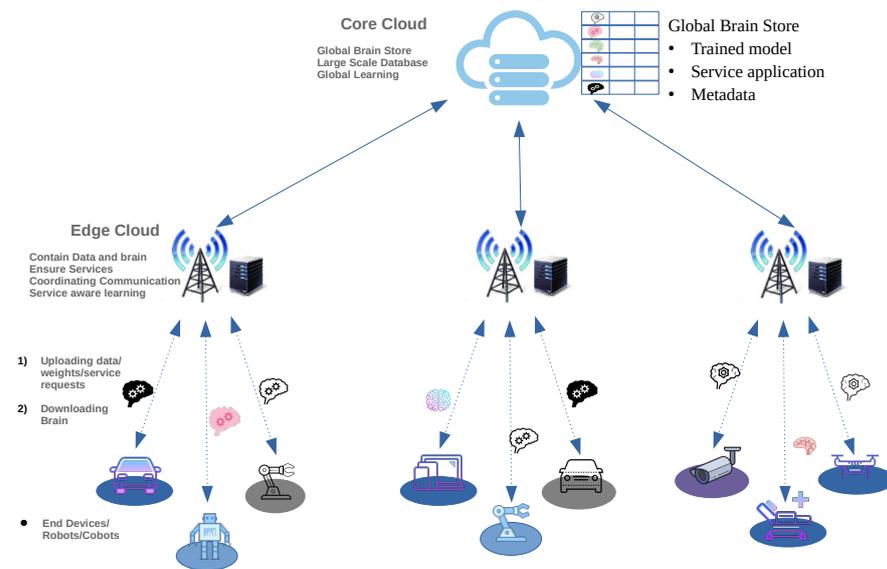


Figure 1. Proposed Brainware Platform.

2. Need for Brainware Computing

Converging all the technologies into robotics outclasses the capability of the performance. Although recent state-of-the-art data processing, artificial intelligence, and task offloading in the edge and cloud computing allow existing robots to perform better in the fixed-domain aspect, it has rather poor performance in multi-domain perspectives. Moreover, the automation industry demands the improvement of ergonomic and productivity issues to enhance the scalability and flexibility of the robots. These constraints lead the automation industry to own a multi-domain aspected computing infrastructure with brain abstraction capability. Brainware Computing infrastructure has been proposed to solve the concerned issues with high potentiality. In this section, we explain the limitations of the related systems with a possible solution and the benefits of Brainware Computing.

However, a multitask model of a particular task can be tuned for another task, but those tasks should be related and from the same domain of shared representation [23]. Multi-task learning strategy increases efficiency by improving generalization through sharing low-level information of same data representation. This is a technique for inductive transfer by using the domain information of related tasks, which is hidden in the dataset as an inductive bias [24,25]. Although a multitask trained and tuned model is able to perform on multiple tasks, the representation and characteristics of the dataset should be similar, i.e., same domain [24]. Multi-task learning failed in the case of heterogeneous data representation of different domains [23]. For example, instead of having individual models for each related task of object detection, semantic segmentation, background subtraction, classification, etc., in images, there may be a single multitask model that can perform on all the tasks using the same data representation by inductive transfer strategy. This model will not work for other different data representations of different domains. However, to improve this limitation and develop a single multi-domain system that is able to adopt and perform better in diverse domains, in this article we propose Brainware Computing. Brainware Computing provides computing infrastructure, which is able to handle and run multiple models for diverse domains and tasks. Multitask learning is one of the aspects in our proposed infrastructure. Based on the service types, both single and multitask learning models are stored in edge and core clouds as the brain images.

2.1. Limitations of Related Systems

Existing robotic systems struggle in terms of performances in multi-domain service aspects due to the lack of variability in the case of multi-domain expertness. Though cloud and edge computing allow high capacity for data processing and data transmission

through 5G high-speed connectivity, recent state-of-the-art robots and cobots are not efficient enough to utilize these benefits. In this section, we elaborate on the limitations of existing robotic systems with possible solutions.

2.1.1. Invariability

In the Robot as a Service (RaaS) [16] infrastructure, usually the service robot is designed for the multi-domain perspectives, but in limited varieties in terms of domain adaptability, even after demanding external maintenance to perform as a multi-domain expert. The rapid rise of advanced robotics technology has opened many new areas. A robot interacts with the environment based on the predefined tasks and a fixed-domain artificial intelligent model in it. If the environment demands a new service, then the robot failed to interact with the environment. Although recently advanced robots update their learning through deep reinforcement learning, they have insufficient capacity to accommodate all the services due to the lack of computational and storage capability. For example, a robot that is designed for cleaning the house in the afternoon fails to perform as a child-caring robot or teacher at night and vice versa. Deploying and enabling different service-oriented brains at a different time in the robot is one of the best solutions to make the robot perform in multi-domain perspectives.

2.1.2. Inefficiency

Recent advancements of the 5G internet, edge, and cloud computing allow high-speed data transmission and processing toward a very large scale [26,27]. Though existing service-oriented architectures such as IaaS, SaaS, and RaaS utilize the data processing advantages of cloud computing, they overlook the possibilities of edge computing and 5G internet [13,14,16]. For example, RaaS provides services based on cloud computing and external control by the developers without utilizing edge computing advancement and high data transmission opportunity [16]. Let us consider a robot that is designed to recognize human action using a deep learning model in videos. If 3D-Resnet101 [28] is selected as the backbone network for this task, then the trained model size will be 365.1 MB considering 64 frames per clip [29]. 5G internet allows a data transmission rate up to 1+ Gbps in peak and 100+ Mbps on average with very low latency [30]. So, on average, recent high-speed 5G internet requires approximately 3 to 4 s to deploy the brain image of action recognition tasks in the robot. Moreover, considering task processing and data sharing capacity in edge computing, this latency can be reduced with improved reliability [31]. Existing robotics technology including service-oriented structures is not capable of grabbing these benefits yet. This scope of high data transmission and computational capacity in edge and cloud computing allows deploying and enabling different service-oriented brains at a different time into the robot to open up a new era in robotics technology. These brains are stored in core and edge clouds.

2.1.3. Lack of Flexibility

The collaborative Robot (COBOT) is a more flexible and scalable robot than classical robots [2]. Cobots interact with the shared environment through collaborating with the human. To solve the ergonomic and productivity issues of the robots, cobots create smooth and strong virtual surfaces in the shared workspace. Cobot improves flexibility, relying on the human intelligence and demanding external control. This dependencies make the cobot flexible, but in a limited scale. Recently advanced robots are more adaptive and reliable than previous ones, but they are less flexible in terms of multi-domain expertness [32]. Both robot and cobot are less flexible in the case of performance in a diverse domain. To enhance the flexibility of shifting intelligence and make the robot perform in multi-domain task spaces as an expert, deploying a domain-specific brain during the execution and removing allows the robot to be highly flexible.

2.2. Benefits of Brainware Computing

Existing AI enabled services have many limitations for better services in the future such as invariability, inefficiency and lack of flexibility. Therefore, Brainware Computing paradigm may solve current limitations furthermore by having the following benefits:

- **Multi domain adaption:** Brainware Computing allows the robots to have a different brain at a different time based on the service demands. As it is not possible to develop a brain in an AI agent that can perform as an expert in multiple domains, changing the brain based on the services is one of the best solutions regarding this issue. The existing related robot does not allow huge computation and storage capacity, so keeping multiple brains and switching between the trained model is not feasible in concept. For example, if a model is trained for language translation using natural language processing principles, then that model is not able to perform using video or image data. Some recent techniques can be adaptive at a multi-domain workspace by sacrificing performances. Domain-specific trained models perform better due to having the easy way of generalization and pattern recognition in organized data instances. As Brainware Computing enables a service-oriented brain based on the service demands and introduces the Brain as a Service (BaaS) infrastructure, the robots can have extensive performances in any corresponding domain at a particular time by shifting the brain.
- **Efficient utilization:** Having a huge computational and storage capacity of cloud and edge computing with high-speed 5G connectivity, proposed Brainware Computing supports virtualization and distribution among the edges and then robots based on the demands efficiently and effectively. Existing RaaS [16] utilizes cloud computing and overlooks the possibilities of edge computing and high-speed connectivity. Brainware Computing allows users to have a more specific, expert, and high-performing brain for the task that is trained using huge datasets and computation facilities. All the global models are stored in the intelligent brain store of the core cloud, then subsets of the whole number of brains, are distributed among the edges based on the local interests. Edges enable, install and remove the brain to/from the robot based on the services. Thus, Brainware Computing pushes up the margin of possibilities of brain re-usability and platform translation for the developers and performs efficient utilization of the available resources.
- **Performance enhancement:** An AI agent or a trained model of general-purpose suffers from poor performance due to the heterogeneous data characteristics and patterns of training data that came from multiple domains. Having a domain-specific and specialized trained model supports the AI agent for performing better. Instead of focusing on a single giant model that is trained in cross-domain nature, Brainware Computing focuses on having domain-specific multiple brains for each service. Hence, robots do not suffer from poor performances due to the data distribution and structure problem of multi-domain datasets during training.
- **Fast new service deployment:** High-speed 5G connectivity allows 1+ Gbps bandwidth in best cases and 100+ Mbps in average cases. Besides Brainware Computing grabs the edge computing as the backbone for the distribution of the data and brain. Unlike RaaS [16], Brainware Computing deploys the brain into the robot from the nearest edges and without demanding explicit command by the administrator or programmer. These autonomous arrangements allow Brainware Computing fast response and deployment of the corresponding brain into the robot.
- **Cost effectiveness:** The intelligent brain store of cloud contains all of the global service-specific brain. The subsets of the brain are shared with the corresponding edges based on the local interests. A single brain can perform at different regions, in different robots and at different times. Providing the control of deploying and removing the brain to the coordinating communication unit of the edges, and the interaction between edges during the searching brain allows the Brainware Computing for reusing and sharing the same brain at multiple times and multiple events.

Moreover, the core concept is to replace all possible sensors by camera and microphone. Advanced emerging computer vision and time series signal analysis techniques are used to perceive the environment and surroundings. The robots will cooperate with humans using vision and voice command. So, instead of using numerous sensors, deep reinforcement learning models are able to perceive the real world using the camera and audio sensors. Suggested general purpose robots contain camera and sound sensors. The task and services are performed using the image and audio data. Proposed Brainware Computing infrastructure provides only those services that are able to perform using images, videos, audio signals data and the processed information obtained by processing them. Utilizing and processing these data, robots estimate and augment other possible information for understanding nature. For example, Minakshi et al. [33] proposed the idea of rainfall prediction by analyzing cloud images. Nilay et al. [34] proposed a technique for weather forecasting using satellite images by avoiding other sensors. Being motivated by these works, we suggest replacing the other sensors by camera and audio sensors. For any particular task, the dedicated and specified corresponding brain images are responsible to extract the necessary environmental information from the images, videos and audio data. So, both the cost of sensors and storage is reduced in the Brainware Computing infrastructure.

3. CONCEPTS: What Is Brainware Computing

Brainware Computing denotes searching, deploying, and enabling brain images to be performed at the robots from edges and cloud, extensive global training, sharing weights, inference optimization, shifting control of brains in robots, and sharing knowledge among the edges. Here, we define any artificial intelligence (mostly deep learning) model that is obtained through supervised, unsupervised, or reinforcement learning and trained in a centralized, decentralized, or federated manner with driving application and metadata as the brain. Initially, the model in the brain images is obtained by training in a centralized manner on a large dataset at the core cloud with high computational capacity. Then, based on the services and demands by the robots, the model updates its learning at a service-aware learning unit in the edge. For this purpose, the robots should contain all possible sensors and actuators with flexible hardware capability. The general purpose robot contains all possible hardware arrangements to interact with the real-world environment regardless of services. During the execution, necessary sensors and actuators among all the components are mapped, equipped, and driven by the service-oriented service application of the brain using metadata.

Figure 1 illustrates the concept of Brainware Computing. Brainware Computing leverages the computation, storage, and data transmission capability of both edge and cloud computing. Core cloud stores all the necessary brains in the global intelligent brain store. Every edge cloud also contains the group of brains in a local intelligent brain store for enabling requested brain images at different times into the robot. Depending on the service demands that are requested by the robots, the edge cloud enables and sends the brain image to that particular robot. For a particular service, the edge sends the corresponding brain image to the robot. After performing the task, the robot removes the brain image (see Figure 2). For example, a general-purpose household robot receives the commands from the house owner through voice recognition for cleaning the house in the afternoon. The robot sends the metadata, which includes service details with surrounding environmental data, to the nearest edge. The corresponding edge analyzes the metadata and environment information for searching for an appropriate brain among the edges and core cloud. The selected brain is enabled and deployed into that particular robot. After finishing the cleaning task, if the same robot is requested to perform the task of taking care of a child, then the previous cleaning brain is disabled and removed from the brain. At night, the same robot can install the surveillance and guard brain image to ensure security by roaming around the house. For performing the new task, the same procedure repeats as described, and the robot performs the requested task. All the robots for all the services follow the same

operational flow as described. The overall flow of execution of the Brainware Computing can be divided into the following concerns:

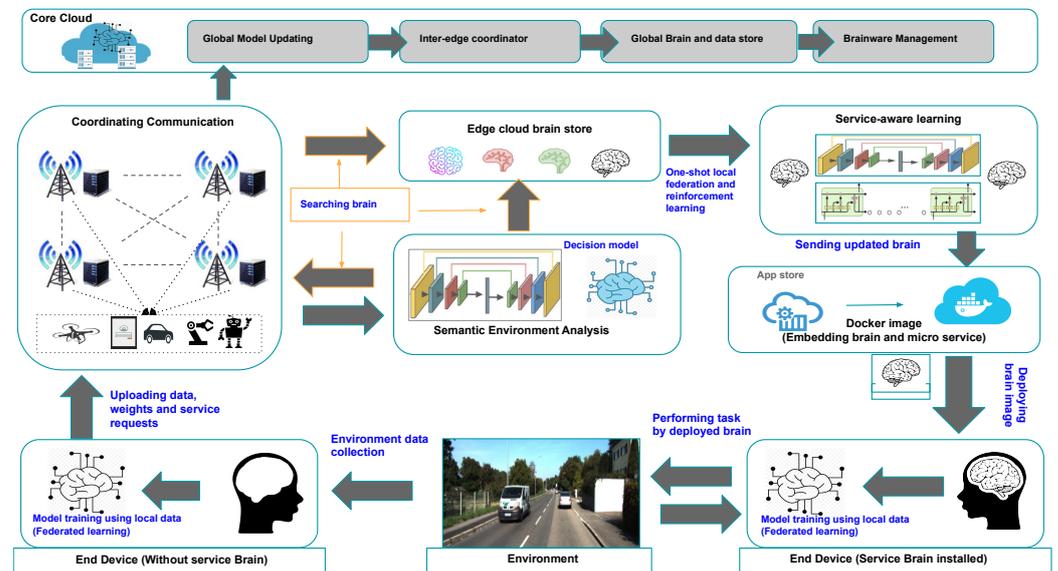


Figure 2. A single operational flow between edge devices and edge cloud with architecture.

1. End devices (Robots, Cobots, etc.)
 - Local end device associative standalone operating application for handling the processing of metadata;
 - Sensors and actuators arrangement.
2. Edge cloud
 - Inter edge coordinating communication;
 - Semantic environment understanding;
 - Inter-edge interfacing for searching for service-oriented brain;
 - Service aware knowledge updating of the enabled brain;
 - Federated learning (Supervised, Unsupervised, Semi Supervised);
 - Reinforcement learning;
 - Encapsulating AI model, software and metadata and managing images;
 - Local edge intelligent brain store;
 - Resources management.
3. Core Cloud
 - Global model updating;
 - Federated averaging;
 - Centralized deep learning;
 - Global intelligent brain store;
 - Inter-edge coordinator;
 - Container management.

The functional impact of the concerned units are elaborately described in the following section.

3.1. BaaS: Service Oriented Brain as a Service in Brainware Computing

The core concept of Brainware Computing is enabling Service Oriented Brain as a Service (BaaS), which allows the existing computing infrastructure and robotics technologies obtaining multi-domain expertise for diverse services without demanding explicit control. See Figure 3 for further details.

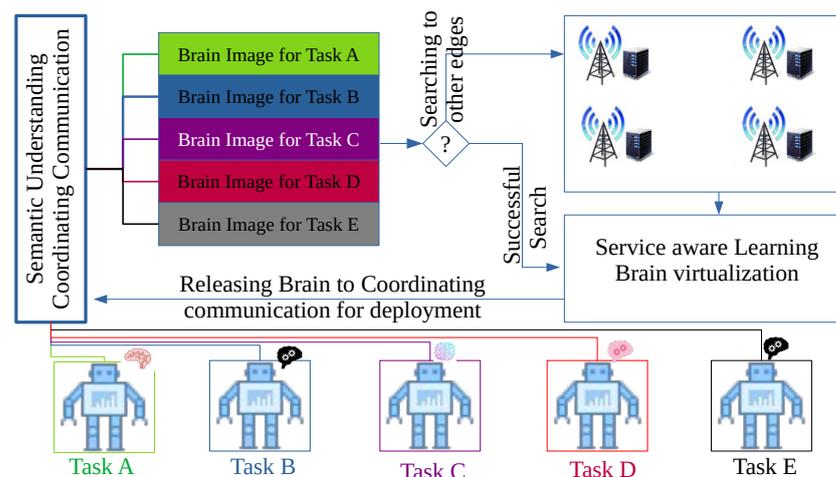


Figure 3. Service-oriented brain searching, learning updates and deployment. (Different colors represent different services).

In 2010, Yinong Chen et al. proposed Robot as a Service (Raas) [16] based on SOA architecture. These service-oriented architectures and robots are equipped to perform multi-domain tasks, but in the fixed domain at a time. RaaS robots demand explicit programming by the developers to make them experts for multi-domain service spaces. In our Brainware Computing, BaaS does not require explicit programming. The whole end-to-end procedure for enabling, deploying, and removing brains inside the robots is done by edge cloud. A general-purpose robot with all the necessary hardware arrangements can perform any of the short and long-term tasks. Recently, a robot can perform multiple tasks such as cops [35], waiters [36], pets [37], child carer [38], autonomous car [39], etc., but it needs to be explicitly programmed, and the flexibility of shifting domain is limited. In Brainware Computing, a general-purpose robot can do all the necessary tasks with high flexibility in terms of domain adaptations such as a car, cleaning robot, pet, UAV, etc. Brainware Computing offers the complete architecture of Service Oriented Brain (SOB). Based on the request of a service broker, the robot requests the nearest edge cloud for the service-specific brain in the form of a virtual machine including the corresponding service application. Edge cloud contains the most possible brains appropriate for corresponding services, for example, brain for UAV, car, pets, cleaning, cops, surveillance, etc. After having the requests from the service broker, the general-purpose robots convey the service request with metadata to the coordinating communication unit of the responsible edge. After processing the service demand, searching, and updating the brain, the coordinating communication unit transmits the requested brain image as a service to that particular robot.

3.2. Brain Virtualization and Service Deployment

Since core and edge cloud can perform large-scale training and contain a huge storage capacity compared to end devices, Brainware Computing focuses on training global artificial intelligent (specifically, deep learning) models in the core cloud. Edge and cloud collaboratively perform federated and reinforcement learning and averaging weights. Trained global service-oriented brains are stored in the intelligent brain store of the core cloud. Each edge cloud contains necessary brains based on the local interests. After having service requests by the robot, the coordinating communication unit sends the request and metadata to the semantic environment analysis unit. The semantic environment analysis unit determines the brain category and searches for the corresponding brain image. Initially, robots just have the operating application which drives them to install and remove the container images. Based on the request types from the robots, inter-edge coordinating communication unit in edge cloud decodes and analyzes the request and search for specific brain image among the edges and often in the cloud. Service-aware learning unit updates the weights of the selected brain using the environmental data sent from the robot using

federated or reinforcement learning. Here, learning is performed considering some key factors such as service category, duration, scales, etc. The updated model is sent to the cloud for globalization using the large dataset by performing averaging weights. The final global model is shared with the corresponding edge brain stores. The selected brain is transmitted to the robot in the form of a virtual machine. Robots can deploy the service-oriented virtual machine on request. The container management unit encapsulates the updated brain, driving application, and metadata to a container image. The inter-edge coordinator unit of the edge cloud is responsible to deploy the service brain to the robot. After performing the task, the brain is removed from the robot by the operating application in the robot.

3.3. Heterogeneous Service Spaces and Task-Space Driven Intelligence Control

In the existing automation systems, the artificial intelligent model in the robot can perform better in a single domain. The operating software in the robot drives its sensors/actuators to meet a specific goal. So, service spaces for the specific robots are predefined. No artificial intelligent model can perform better in other domains after being trained. In our proposed Brainware Computing, the key concern is Service-oriented Brain as a Service (SOB). In this system, each robot can handle different brains at different times. A single robot can lead itself to perform accordingly for any kind of services. With the ability to install and remove different kinds of service-oriented brain, our proposed robot structure can work at heterogeneous service spaces, for instance, self-driving car, cleaning, language translation, etc. Based on the service demands, any kind of service can be performed by the robots by enabling the specific brain from the edge. To control the task-driven intelligence and switch from one intelligence to another, the end device robot contains a controller operating application. Furthermore, there is the concern of utilizing and synchronizing with the same arrangements of sensors and actuators. The robots contain cameras and audio sensors to receive environment information from the real world environment. The brain images that contain the service application and the controller application in the end devices are synchronized together to interact with the real-world environment.

3.4. Service Request Analysis: Semantic Environment and Metadata Understanding

Inter-edge coordinating communication blocks also receive service requests, images, and time-series signals from the robot. After performing initial decoding of the command string and environmental data, the semantic environment understanding block processes the surrounding photos and time-series signals to understand the environment. To do so, the deep learning-based model is deployed to resolve the information of “what” and “where”. Deep learning-based encoder–decoder for semantic segmentation, which refers to “what”, and depth estimation for “where” information, which indicates the location of the objects. There is an LSTM-based sub-block to understand the surrounded time-series information for better understanding, see Figure 4.

Using the outcome from the encoder–decoders, the decision model translates that information into a decision string. In this work, we propose three kinds of translators, such as segmented image-to-text translation, depth image-to-text translation, and signal-to-text translation. All these translations contain deep learning-based encoder–decoder for time series data analysis. The outcomes help to determine the decision for searching the brain. This decision model decides which brain the searching block should search, for example, “self-driving car”, “Cleaning”, “Language Translator”, “Surveillance”, etc.

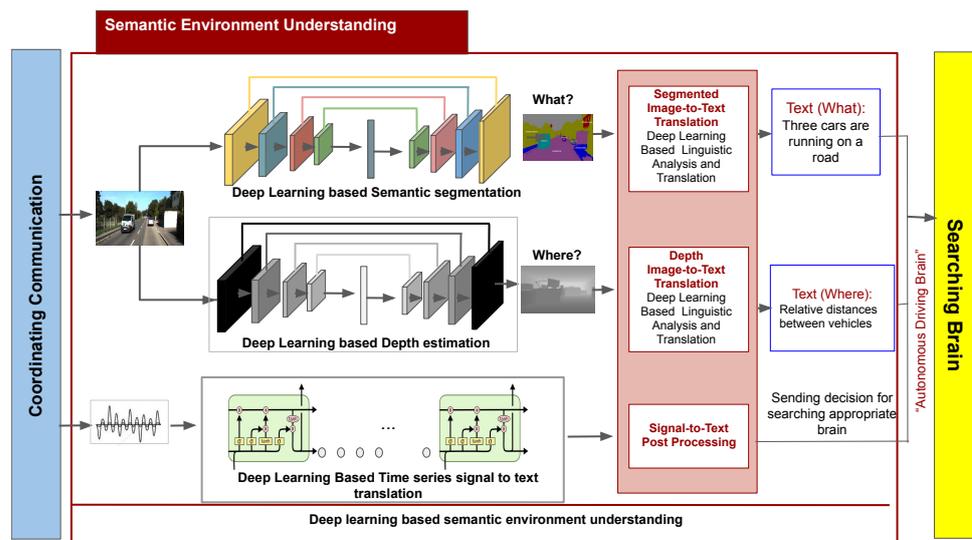


Figure 4. Semantic environment understanding.

3.5. Service Aware Learning

For each complete service cycle, the corresponding edge cloud updates model weights depending on the service specification and current environmental uncertainty. Some services require short response time with simple intelligence where others might demand long response time with complex intelligence.

Few services are the combination of multiple services, but some services are atomic in nature. Depending on the level of intelligence, sensitivity, duration, and the type of services, the service-aware learning unit in the edge cloud updates the model using the current environmental information and metadata (see Figure 5). Based on the services, the learning can be supervised, semi-supervised, unsupervised, reinforcement, a classical or deep learning manner, etc. The technique can be federated or centralized in service-aware learning. For example, if a robot sends a request to an edge for the UAV brain for an intrusion detection and surveillance brain, then the selected brain does not need to update its learning at the service aware learning unit. However, if the robot requests for home cleaning, then the brain needs to be updated using the current scenario based on semantic environmental data. During the execution of the task, the brain is updated using federated learning, and then the cloud computing performs global averaging.

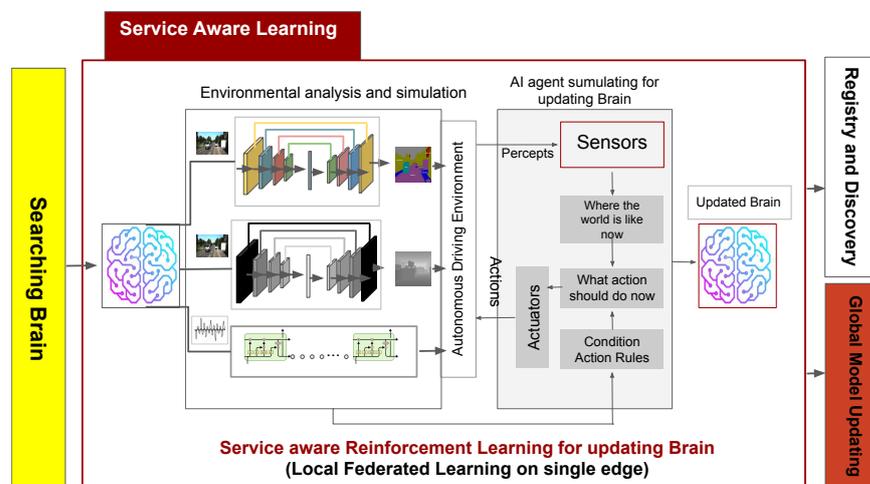


Figure 5. Service aware knowledge updating in the edge cloud.

3.6. Inter Edge Coordinating Communication

Inter edge coordinating communication block handles all sorts of communications between edges, edge-core cloud, and edge-robots. Inter edge coordination block acts as the controller of the Brainware Computing communication. Requests from the robots to edges, controlling communications between edges, the protocol of sharing a brain, sending models for updating to core cloud and receiving data and brain for computing and storing to the brain store are performed by the inter edge coordinating communication unit (see Figure 2).

3.7. Distributed Hierarchical Brain Store

All the trained models and huge datasets are stored in the edge and core cloud, and the brains are shared among the edges based on the local interests and the intensity of service demands. We introduce a concept called Brain Store in the edges and core cloud. The brain store stores the container image of brains (see Figure 3), which contain trained models, metadata, and service applications together with metadata, which guide the robot OS to install and remove the brain from itself. Each brain in the brain store is trained to meet different purposes. For example, if we consider three services such as services A, B, and C, each category has some subcategory that defines a specific task. For instance, A belongs to the self-driving car service category, then A_1 belongs to the self-driving car brain for those countries where the car is driven through the right side, A_2 maybe for others, where it's driven through the left side. A trained AI model is stored based on the statistical interests on the various edges (see Figure 1). For example, if service A is much more frequently used in the location where edge-1 is suited, then brain A will be stored in edge-1. In case of need of brain A by another edge such as edge-2, edge-3, etc., then edge-1 shares that brain with the corresponding edges. This sharing task will be performed by an inter-edge coordinating block. Different kinds of brains are stored in different kinds of edges based on the service interests of the local region.

3.8. Encapsulating Brain Images

As the purpose of Brainware Computing is to deploy different kinds of brains at a different times based on the services in the same robot, we need to be concerned about the collecting and processing of sensors input and acting through the actuators. For various services, we need to enable a particular kind of brain because the input–output relationship of different services is different. For each particular brain, we need to have a particular service application and metadata that can run the brain and lead the robot to interact perfectly with the environment through sensors and actuators. Hence, to run the specific brain, we need to install a specific running service application with specific metadata in the robot together. For this purpose, we encapsulate the service application and metadata with the corresponding trained model (see Figure 3). During the service execution, the service application leads the trained model and performs the task. Encapsulating service application with corresponding trained models and metadata is done by the container management block in the edge cloud after updating the model.

4. Scope and Possibilities

4.1. Indoor Intelligent Robots

Brainware Computing provides Brain as a Service (BaaS) to both kinds of fixed and movable robots and cobots. In recent years, various kinds of robots and cobots are being used to enhance productivity in the industry. Instead of using the fixed intelligent robot, Brainware Computing allow the robots and cobots to have various kinds of the brains in different times based on the requested service such as moving object detection, surveillance systems, caring for a child, cooperation with humans, etc., using the arrangement of the sensors and actuators (sensors are replaced by the cameras and microphones). For example, in a soap factory, the robots will carry the soap materials to the holders, and if needed then remove that brain and install the brain for controlling the machines for converting soap

materials to the soap. Based on the need, that robot can again install the product packaging brain to cooperate with the workers, and again, the moving Brain will be installed to move the soap packets to the soap store for delivery. So, one robot can work, act, and serve in multiple tasks. Furthermore, based on the needs, the edge can install a collaborative brain, which will allow the robot to work with a human, sensing human response, commands and reacting to a shared environment.

A self-driving car is the best example of an autonomous movable robot in the factory to detect, collect and move the product. These kinds of robots need various brains to operate for the various purposes at different times. In industry, nowadays collaboration robots are performing better. Humans and robots are working together at the shared environment. Humans have some limitations in terms of speed, mistakes, tiredness, emotion, etc., and the accuracy of the intelligence systems is not 100% because of diverseness and uncertainty of the real world environment. Combining performances of humans and cobots increased the performance at services. In our proposed technique, Brainware Computing will install the brain in the cobot, which will take commands, observe humans, be a learner of human intelligence and work with humans together for different perspective at different times.

4.2. Outdoor Intelligence

Self-driving cars, surveillance systems and other outdoor intelligent robotic systems need to be improved and demand Brain as a Service (BaaS) infrastructure due to overcome existing rigidity and fixed type intelligent approach. A self-driving car and other outdoor intelligent systems sense the environment and cooperate with humans (if necessary) using the arrangement of sensors and actuators. Using Brainware Computing possibilities, an outdoor robotic system can install and deploy the requested service brains based on the environment state, service categories and time, which leads the robots or cobots to perform better. For example, a robot which is able to interact with humans to drive a car using a driving brain image, can also turn into an expert of a different domain by removing the driving brain and installing that particular service brain (i.e., for cart, surveillance systems, road cleaning robots etc.) and vice versa.

5. Challenges and Opportunities

Section 4 demonstrates the scope and potential case studies of Brainware Computing. Though Brainware Computing opens up a new era in robotics and intelligent computing, there are some challenges to integrate this large scale computing architecture for real-world execution in terms of resources, services, privacy, and security as edge computing also contains some challenges and limitations [40]. In this section, we elaborate some challenges in Brainware Computing with some possible solutions and opportunities, along with further research directions such as resources management, service management, resource optimization, privacy, and security.

5.1. Resources Management

5.1.1. Computing Resources Management

Computing resources management denotes the process of allocating and de-allocating system resources (cpu, memory, io, and cache) for different applications and threads. Computing resources management in Brainware Computing can be divided into three concerns: (1) In cloud, (2) in edge and (3) in robot.

In Brainware Computing at all of the three places, we are motivated by the elastic memory management [41] and virtual memory optimization technique [42]. Recent resource management applications depend on the containers such as Docker [43], Kubernetes [44], or YARN [45] containers. These containers make the applications isolated and share the same machine by providing hardware resource limits. Applications for resource requirements are managed and controlled by the containers. Based on the service, a specific brain image or a collection of related brains is installed at different times. It denotes that, to perform the service for specific tasks, sometimes they need multiple intelligent brains

collaborating with each other. For example, for driving a car, the robot needs a self-driving car brain to collaborate with humans, which consists of multiple sub-module brains such as semantic segmentation brain, motion detection, object detection, language translation brain, etc. In this case, multiple brains collaborate with each other. These modules create a driving model. After performing the task, the driving brain which consists of two modules is removed from the robot. On the other hand, in the case of atomic tasks, for example, a COBOT as the packaging machine, which helps humans for product packaging in a factory. Here, a single computer vision model is installed to accomplish the whole task. So, in terms of hybrid tasks, multiple brains work side-by-side, but for atomic tasks, a single brain works alone. Based on the service, different brains or a collection of corresponding brains are installed and removed from the robot at different times.

In possible scenarios, the necessary brains are installed, and after performing the task, the brains are removed from the robot. Moreover, in some cases, to run a successful execution, there need to run a collection of virtual machines for the same purpose side-by-side, which is controlled by the hypervisor.

5.1.2. Network Management

The coordinating communication unit in edge cloud (see Figure 2) interacts with the core cloud and robot and shares brain images among them seamlessly. This infrastructure turns the networking arrangement complicated by the robots and edges because of heterogeneity, service nature, interaction, density, and transmission protocol. Thus, performing uninterrupted communication demands supports at all the layers in the internet architecture such as network layer, MAC layer, and transport layer, which affects the overall performance of the internet [46]. Brainware Computing focuses both on energy efficiency by providing higher throughput. Sharing brain images among the edges and for the real-time interaction with the robots by the coordinating communication unit are necessary. The whole Brainware Computing architecture performs using hybrid protocol depending on the interaction types. For deploying a brain into a robot, and sharing brain images and metadata among the edges, hybridization of communication protocols supports both low and high-level services, data dissemination, and accumulation perfectly. It also requires constant management of virtual machine abstract layer management, priorities services, queue management, etc.

5.1.3. Operation Management

For the sake of resources management, the coordinating communication unit in the edges manages processes, metadata and threads to interact and share information among the edges, cloud and robots, prioritizing service requests and operation in an efficient manner. In Brainware Computing environment, multiple service requests may happen at the same time. Furthermore, atomic operations of different services in edge and cloud may need to be performed at the same time in the same edge. However, managing these heterogeneities, task sharing processes and operations among the edges, prioritizing activities are handled by the virtual execution model in coordination communication unit and hypervisor. Synchronizing these operations and managing seamless service is quite challenging.

Brainware Computing communication is motivated by service oriented priority-driven execution systems [47] and event-driven scheduling and stack [48]. Coordinating communication units in edges and robot's event handlers continuously wait for the service request from the service broker, metadata sharing, activity and process offloading and internal or external events, such as deploying and removing brain, metadata sharing, environment data analysis, searching for brain, inference optimization, braualization, real time service aware learning, etc. The virtual machine in the edges dynamically allocates the memory stack to the operation and the event handler in the edge performs the run-to-completion task. All the operations and activities share the same process bus-line and utilize limited memory efficiently.

5.1.4. Metadata Sharing and Translation

In Brainware Computing, whole service flows from the service broker request to the completion of the task generates and translates different kinds of metadata. In every brain image, there is a need to manage the metadata. Initially, the service broker requests a service from the nearest robot by sharing descriptive metadata for the discovery and identification of the service type ID, location and other necessary information regarding service specification. The robot processes the metadata shared by the service broker. Translating service requests metadata and encapsulates new administrative metadata, and the robot shares the metadata with the coordinating communication unit of the edge cloud. Along with administrative metadata, which denotes the resources type, service locations, time and type, the edge cloud produces statistical metadata for interacting with other edges and clouds. During enabling the service oriented brains (SOB) and services execution time, metadata in brain images helps drive the application to run the brain. Handling these metadata inside the proposed gigantic architecture demands careful concern.

5.1.5. Hardware Management

Robots interact with the real world environment through sensors and actuators. They perceive various kinds of environmental input data, process them and react to the environment based on the specified task. Among all the input data types, vision and perceiving time series data are quite important. However, different services demand different sensors and actuator arrangements to interact with the robots. For example, a robot perceives a visual scene, time series data, motion, GPS, ultrasonic distance, etc., information for enabling a self-driving car brain, and reacts through wheels, brake, accelerator, etc., where a cleaning robot does not need to examine the GPS and motion of surrounding objects often and react through wheels, motor, arm and brush, also a UAV needs the altitude and 3D camera information to perceive a visual scene and environment and react using motor, fan, monitor, etc. As Brainware Computing allows all the general purpose robots to be experts on multi-domain spaces enabling different brains at different times, the robots only cameras and audio sensors and actuators with specified port ID. All other types of necessary information is extracted from the image, videos and time series data. That means general purpose robots only have the cameras and audio sensors. Service application handles the port and necessary camera and audio sensors and actuators. For example: the general purpose robots have only cameras and audio sensors for receiving images, time series signals, and other information such as motion, 3D scenes, etc. are extracted using the camera and audio information. Hardware configuration in robots should support installing heavy brain images.

5.2. Service Management

To ensure service and intelligent brain management at the edges, we focus on the following four fundamental features. These features need to be taken care of for developing a reliable system, for instance, differentiation, extensibility, isolation, and reliability.

5.2.1. Differentiation

With the increasing impact of robotics in diverse applications and needs for providing heterogeneous services in multiple sectors, robots are demanded and requested to perform multiple services at different times, such as cleaning robots, self-driving cars, UAVs, smart homes, etc. These service requests from multiple service brokers have different priorities and importance. For example, a service request of patient caring robots should be performed earlier than the service requests of cleaning robots.

5.2.2. Extensibility

Purpose of Brainware Computing is to provide any services that are requested by the users gradually. Usually, the existing robotic systems perform on multiple domains through Robot as a Service (RaaS), but demand explicit programming by the robots. As Brainware

Computing contains all the necessary brains for all the services, often the category of services may be increased and a new service may need to be registered to the core and edge cloud. The demands from service broker are unknown and the service spaces are uncertain and unseen. Robots need to increase the number of services and also update the learning. In Brainware Computing, we propose service oriented learning to add new services and also to update the learning of existing models. This registry and discovery task is accomplished by the registry and discovery unit of edge clouds.

5.2.3. Isolation

To ensure the isolation in Brainware Computing, we suggest designing the architecture using virtual machine for each unit of execution and container image of the brains. The overall core and edge cloud is designed based on a virtual machine working principle. All the service brains and applications are contained in the form of container images. On the whole, execution flow for any particular services are performed by all the unit successfully. A well-designed collection and set of virtual machine units share the processing. The processing task of a unit does not depend on other units. As every individual service brain and execution block unit is isolated, each using container images and virtual machine, respectively, so the performance of a service does not depend on the other brain. The execution blocks and brain images share processing capability. If a flow of execution is interrupted in a edge because of the failure of semantic environment understanding unit of that particular edge, then the nearest edge shares the semantic environment understanding unit for processing the task of the failed unit and returns the decision for searching brain to the searching unit of the first edge.

5.2.4. Reliability

There are two kinds of reliability issues in Brainware Computing: (1) Brainware reliability in terms of seamlessly service deployment and (2) robots performance during the execution

- Service flow and deployment: For performing the overall service flow from service broker request to the service deployment is quite challenging. There are a lot of unit operations should be performed. The overall atomic individual and isolated systems should be designed to maintain the whole system. We prefer virtual machine for every unit in an isolation manner, where the controller and inter-edge coordinating communications are able to detect the problems and notify all the other edges, then the controller offloads the task to the nearest edge or direct to the cloud. For example: if a service is requested to edge A, and edge A is failing to operate, then all the units of A can detect the reason and notify the cloud, then cloud shift that particular service request to the nearest edge, which ensures seamless services.
- Robot performance during execution: Robot performance depends on the learning strategy and scale. The performance of the brain is concerned with the user. For example, considering deep learning, as all the service brains are obtained after training on large scale data such as imagenet [49], cifar [50], custom dataset, etc., for image processing, kinetics dataset [51] for action recognition, cityscapes [52], for semantic segmentation, etc., using centralized approach, then the performance of the trained deep learning model is remarkable. Furthermore, knowledge updating using federated and reinforcement learning is performed by the service aware learning to ensure it is adaptive to uncertain scenarios.

5.3. Optimization

In Brainware Computing, there are different labels of execution such as, at service broker, robots, edge, and cloud, which are divided into multiple layers. These levels and layers perform different kinds of computation. In this proposed computing, request allocation, task offloading and synchronization are the core concerns. We need to design the infrastructure by specifying the computation and task processing at different levels

and layers. In the case of service request execution, processing, service management, and deployment, there are some key factors that need to be concerned and optimized. Based on the services, we need to focus on the trade-off. Some services require the speedy deployment of the brain images, utilization of energy for efficient and accurate services by interacting with the environment, and low latency using less bandwidth and energy.

5.3.1. Latency

There are some basic kinds of interactions in Brainware Computing, (1) robot-environment interactions and (2) robot-edge interactions (3) inter-edge interactions and (4) edge-cloud interactions. Considering performances, latency is one of the key metrics to evaluate the quality of service (QoS) and the response time of service requests. As the core and edge cloud have the high computational capability and the artificial intelligence model parameter estimation task is done by the edges and core cloud during training, there is less computation time needed. By the grace of 5G internet, the transmission delay has drastically reduced. However, deploying or sharing the brains among the edges and from cloud to edges and computation for searching and preparing brain images may add some latency. As the core and edge cloud have a high computational capacity, high-speed networking reduced the latency. In Brainware Computing, robots interact with the real-world environment, being driven by the brain image. As the brain model can be heavy (mostly, deep learning model), robots only perform inference for reacting to the environment using the model. All the necessary training and model updating is performed on core and edge clouds, respectively. Interaction between edges also has an issue of latency, but because of high-speed connectivity among cloud and edges and computation power, this problem can be solved. The latency is handled by dividing the computation into different layers. For speedy interaction with the real-world environment, Brainware Computing imposes the majority of the workload on the cloud and edges, where robots only infer using the trained model.

5.3.2. Bandwidth

To share and deploy brain images into robotics with high bandwidth can reduce transmission delay, especially for a heavy deep learning model. As in Brainware Computing, the service requested may arrive from remote places where internet connectivity is not so smooth and seamless, the robot first deploys the whole brain into it, then performs offline. At service execution time, no bandwidth is required. While deploying the brain, the transmission latency depends on bandwidth. Lower bandwidth increases transmission delay. In Brainware Computing, we prefer a trade-off. As all the workloads are performed in the core and edge cloud, there is less transmission delay required. The bandwidth between the core and edge cloud or edge cloud and robot is considered and tends to be high for sharing the heavy model during deploying and sharing weights. We prefer model optimization and encoding–decoding technique for sharing brain image to reduce the bandwidth required and transmission delay. As in most cases, the distances between robots and edges are uncertain, then to reduce bandwidth, we prefer to compress the trained model and transmit using a packet system. However, we focus on bandwidth and latency for training in the cloud, updating the model in edges and inference only on robots and sharing the brain using compression and decoding in different layers.

5.3.3. Energy

The robot performs using battery energy, which is the most important resource. To save the robot's battery energy, we take away the whole workload of training and learning updating (federated and reinforcement learning) at the core and edge cloud. The robot's layer only shares the metadata, installed brain images and inference using the trained model using low energy compared to the related systems. The approach for installing the brain and performing only inference reduces the energy consumption by the robot

comparatively. We provide most of the computation workloads to the edges and cloud to save energy of the robot's battery.

5.4. Privacy and Security

In the proposed Brainware computing architecture, robots collect and share data with the nearest edges. Inter-edge communication needs to share data together. Sometimes, robots may collect and send sensitive information, which is the fundamental concern of privacy.

If a robot is set for service in a home, office or any other confidential place, a lot of private information can be collected by the cameras and microphones. In this case, Brainware Computing performs federated learning to keep sensitive data confidential.

To ensure the privacy, we suggest two approaches for two different purposes: one is federated learning [53] for training using local sensitive data, and the second one is a deep learning encoder–decoder [54] approach (see Figure 6) to ensure end-to-end encryption and decryption facilities in all the sender and receiver end during data sharing among the edges. According to the approach, during data sharing, each robot encodes data using the encoder part of the model and shares only the encoded features with the nearest edges. Coordinating communication blocks includes the decoder of the encoder–decoder network and decodes the encoded features, which has been sent by the robots. Each edge holds data privacy using the encoder–decoder approach. However, in Brainware Computing systems, only the encoded features of all kinds of data such as images, voices, metadata etc., are shared to ensure privacy. More specifically, images and voice data are encrypted using convolutional and recurrent neural network-based encoder–decoders because of their better performance. Moreover, in terms of global learning using local sensitive data, Brainware Computing uses federated learning [53] strategy where the local model in the robot learns from the local data. Global learning blocks in cloud computing perform global federated averaging to obtain a globally generalized trained model.

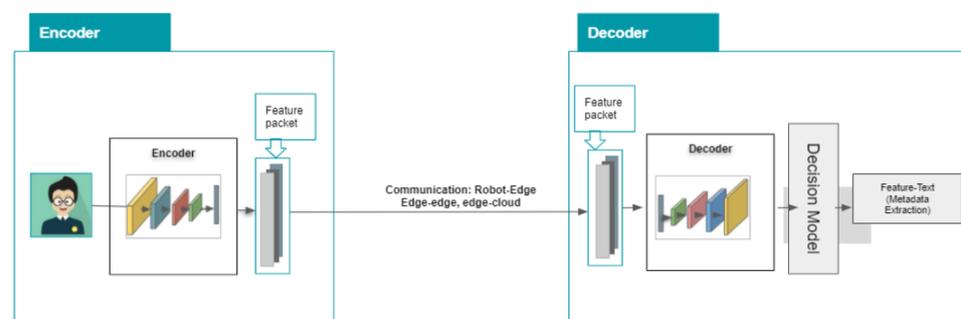


Figure 6. Encoder–decoder example for transferring the encoded features of images instead of actual data.

For the purpose of security, there is a section called registry and discovery in the edge clouds. The following concern should be maintained to ensure security for the Brainware Computing. Saad Khan et al. [55] and Shalin Parikh et al. [56] introduced some techniques to ensure the security for fog and edge computing, respectively. Being motivated by these concepts, in Brainware Computing, we suggest the following steps:

- Federated learning to hold the data privacy in edge computing;
- Encryption of metadata and access verification and authorization: Registry and discovery block is responsible for this task;
- To detect unauthorized access the block includes intrusion detection systems (IDS) and multiple step verification systems;
- Analysis and identification of User Behaviour Profiling (UBP).

6. Conclusions

Recently, numerous service-oriented infrastructures are being proposed with the help of edge and cloud computing, which ensures strong connectivity, data processing and sharing with low latency and high reliability. Existing robotics technology cannot properly utilize the whole resources, on the other hand, existing robots are not multi-domain experts. Though Robot as Service (RaaS) provides some flexibility and programmability, it contains no edges, hence it requires high latency and demands external programming by the developers. By the advancements of edge and cloud computing, the border of possibilities of the robotics technology has been pushed up by Brainware Computing, which allows the robots to be experts on the multi-domain systems using the different brains at different times without any external coding required by the developers, and needs low latency and bandwidth. We proposed a computing infrastructure, and addressed Brainware Computing with scopes and possibilities. Furthermore, we provided an extensive explanation of the challenges and opportunities with resources optimization. In the introduction of Brainware Computing, we extensively explained every unit, which indicates a new research direction. Furthermore, we experienced that the challenges and opportunities discussed in this paper indicate a new research area. We hope that the proposed Brainware Computing will revolutionize the automation industry with the help of Brain as a Service (BaaS).

Author Contributions: Conceptualization: E.-N.H.; Supervision: E.-N.H.; Writing—original draft: E.-N.H. and M.I.H.; Writing—review and editing: E.-N.H. and M.H.I. Both authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2021-2015-0-00742) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from both the first and corresponding author.

Acknowledgments: A lot of thanks goes to the colleagues for effective contributions to this proposal. The authors appreciate the contribution of the editors and reviewers for their constructive suggestions and insightful comments.

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

References

1. Perrey, R.; Lycett, M. Service-oriented architecture. In Proceedings of the Symposium on Applications and the Internet Workshops, Orlando, FL, USA, 27–31 January 2003; pp. 116–119.
2. Peshkin, M.A.; Colgate, J.E.; Wannasuphprasit, W.; Moore, C.A.; Gillespie, R.B.; Akella, P. Cobot architecture. *IEEE Trans. Robot. Autom.* **2001**, *17*, 377–390. [\[CrossRef\]](#)
3. Butter, M.; Rensma, A.; Kalisingh, S.; Schoone, M.; Leis, M.; Gelderblom, G.J.; Korhonen, I. *Robotics for Healthcare*; European Commission EC: Brussels, Belgium, 2008.
4. Dario, P.; Guglielmelli, E.; Allotta, B. Robotics in medicine. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94), Munich, Germany, 1 January 1994; Volume 2, pp. 739–752.
5. Taylor, R.H.; Kazanzides, P.; Fischer, G.S.; Simaan, N. Medical robotics and computer-integrated interventional medicine. *Biomed. Inf. Technol.* **2008**, *73*, 617–672.
6. Demir, S.; Paksoy, T. AI, Robotics and Autonomous Systems in SCM. In *Logistics 4.0: Digital Transformation of Supply Chain Management*; CRC Press: Boca Raton, FL, USA, 2020; p. 156.
7. Ahmed, I.; Din, S.; Jeon, G.; Piccialli, F.; Fortino, G. Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning. *IEEE/CAA J. Autom. Sin.* **2020**, *8*, 1253–1270. [\[CrossRef\]](#)
8. Macrorie, R.; Marvin, S.; While, A. Robotics and automation in the city: A research agenda. *Urban Geogr.* **2020**, *42*, 1–21. [\[CrossRef\]](#)
9. Golubchikov, O.; Thornbush, M. Artificial Intelligence and Robotics in Smart City Strategies and Planned Smart Development. *Smart Cities* **2020**, *3*, 1133–1144. [\[CrossRef\]](#)
10. Petrlík, M.; Báča, T.; Heřt, D.; Vrba, M.; Krajník, T.; Saska, M. A robust uav system for operations in a constrained environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2169–2176. [\[CrossRef\]](#)

11. Bhatt, P.M.; Malhan, R.K.; Shembekar, A.V.; Yoon, Y.J.; Gupta, S.K. Expanding capabilities of additive manufacturing through use of robotics technologies: A survey. *Addit. Manuf.* **2020**, *31*, 100933. [CrossRef]
12. Grimbale, M.J.; Majecki, P. Nonlinear Automotive, Aerospace, Marine and Robotics Applications. In *Nonlinear Industrial Control Systems*; Springer: London, UK, 2020; pp. 699–759.
13. Dubey, A.; Wagle, D. Delivering software as a service. *Mckinsey Q.* **2007**, *2007*, 6.
14. Dawoud, W.; Takouna, I.; Meinel, C. Infrastructure as a service security: Challenges and solutions. In Proceedings of the 7th International Conference on Informatics and Systems (INFOS), Cairo, Egypt, 28–30 March 2010; pp. 1–8.
15. Keller, E.; Rexford, J. The “Platform as a Service” Model for Networking. *INM/WREN* **2010**, *10*, 95–108.
16. Chen, Y.; Du, Z.; Garcia-Acosta, M. Robot as a service in cloud computing. In Proceedings of the Fifth IEEE International Symposium on Service Oriented System Engineering, Nanjing, China, 4–5 June 2010; pp. 151–158.
17. Kurfess, T.R. (Ed.) *Robotics and Automation Handbook*; CRC Press: Boca Raton, FL, USA, 2018.
18. Who Did Actually Invent the Word “Robot” and What Does it Mean? Available online: <https://www.roboticsacademy.com.au/who-invented-the-word-robot-and-what-does-it-mean/> (accessed on 22 March 2021).
19. Clarke, R. Asimov’s laws of robotics: Implications for information technology. 2. *Computer* **1994**, *27*, 57–66. [CrossRef]
20. Shi, Y.; Wang, N.; Zheng, J.; Zhang, Y.; Yi, S.; Luo, W.; Sycara, K. Adaptive Informative Sampling with Environment Partitioning for Heterogeneous Multi-Robot Systems. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24–30 October 2020; pp. 11718–11723.
21. Buehler, J. Capabilities in heterogeneous multi-robot systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22 July 2012; Volume 26, No. 1.
22. McKerrrow, P.J.; McKerrrow, P. *Introduction to Robotics*; Addison-Wesley: Sydney, Australia, 1991; Volume 3.
23. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
24. Zhang, Y.; Yang, Q. A survey on multi-task learning. *IEEE Trans. Knowl. Data Eng.* **2021**. [CrossRef]
25. Sener, O.; Koltun, V. Multi-task learning as multi-objective optimization. *arXiv* **2018**, arXiv:1810.04650.
26. De Looper, C. What Is 5G? The Next-Generation Network Explained. *Digital Trends*. 5 May 2020. Available online: <https://www.digitaltrends.com/mobile/what-is-5g/> (accessed on 6 June 2021).
27. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A. D.; Katz, R.; Konwinski, A.; Zaharia, M. A view of cloud computing. *Commun. ACM* **2010**, *53*, 50–58. [CrossRef]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. Crasto, N.; Weinzaepfel, P.; Alahari, K.; Schmid, C. Mars: Motion-augmented rgb stream for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7882–7891.
30. Gopal, B.G.; Kuppusamy, P.G. A comparative study on 4G and 5G technology for wireless applications. *IOSR J. Electron. Commun. Eng.* **2015**, *10*, 67–72.
31. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
32. Zhang, Y.; Lu, M. A review of recent advancements in soft and flexible robots for medical applications. *Int. J. Med. Robot. Comput. Assist. Surg.* **2020**, *16*, e2096. [CrossRef]
33. Gogoi, M.; Devi, G. Cloud Image Analysis for Rainfall Prediction: A Survey. *Adv. Res. Electr. Electron. Eng.* **2015**, *2*, 13–17.
34. PKapadia, N.S.; Rana, D.P.; Parikh, U. Weather Forecasting using Satellite Image Processing and Artificial Neural Networks. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 1069.
35. Wired Blog, Robot Cops to Patrol Korean Streets. 17 January 2006. Available online: <https://www.wired.com/2006/01/robot-cops-to-p/> (accessed on 22 March 2021).
36. Cheong, A.; Lau, M.W.S.; Foo, E.; Hedley, J.; Bo, J.W. Development of a robotic waiter system. *IFAC-PapersOnLine* **2016**, *49*, 681–686. [CrossRef]
37. Robot Pets. Available online: <http://en.wikipedia.org/wiki/AIBO> (accessed on 22 March 2021).
38. Ian Hamilton, Robot to be Added at Hoag Hospital Irvine. *InTouch News*, 8 October 2009. Available online: <http://www.intouchhealth.com/> (accessed on 22 May 2021).
39. Liu, S.; Liu, L.; Tang, J.; Yu, B.; Wang, Y.; Shi, W. Edge Computing for Autonomous Driving: Opportunities and Challenges. *Proc. IEEE* **2019**, *7*, 1697–1716. [CrossRef]
40. 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.
41. Wang, J.; Balazinska, M. Elastic Memory Management for Cloud Data Analytics. In Proceedings of the 2017 USENIX Annual Technical Conference (USENIXATC 17), Santa Clara, CA, USA, 12–14 July 2017; pp. 745–758.
42. Deshmukh, P.P.; Amdani, S.Y. Virtual Memory Optimization Techniques in Cloud Computing. In Proceedings of the 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), San Salvador, El Salvador, 22–24 August 2018; pp. 1–4.
43. Docker Container. Available online: <https://www.docker.com/> (accessed on 25 February 2021).
44. Kubernetes. Available online: <http://kubernetes.io/> (accessed on 25 February 2021).

45. Vavilapalli, V.K.; Murthy, A.C.; Douglas, C.; Agarwal, S.; Konar, M.; Evans, R.; Graves, T.; Lowe, J.; Shah, H.; Seth, S.; et al. Apache Hadoop YARN: Yet another resource negotiator. In Proceedings of the 4th Annual Symposium on Cloud Computing (SOCC'13), New York, NY, USA, 1–3 October 2013; pp. 5:1–5:16.
46. Forouzan, B.A. *TCP/IP Protocol Suite*; McGraw-Hill Higher Education: New York, NY, USA, 2002.
47. López, J.M.; Díaz, J.L.; Entrialgo, J.; García, D. Stochastic analysis of real-time systems under preemptive priority-driven scheduling. *Real-Time Syst.* **2008**, *40*, 180. [[CrossRef](#)]
48. Jang, J.; Jung, J.; Cho, Y.; Choi, S.; Shin, S.Y. Design of a lightweight TCP/IP protocol stack with an event-driven scheduler. *J. Inf. Sci. Eng.* **2012**, *28*, 1059–1071.
49. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
50. Krizhevsky, A.; Hinton, G. Convolutional deep belief networks on cifar-10. Unpublished Manuscript. 2010, Volume 40, pp. 1–9. Available online: <https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf> (accessed on 21 March 2021).
51. Carreira, J.; Zisserman, A. Quo vadis, action recognition? A new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
52. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
53. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [[CrossRef](#)]
54. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)]
55. Khan, S.; Parkinson, S.; Qin, Y. Fog computing security: A review of current applications and security solutions. *J. Cloud Comput.* **2017**, *6*, 1–22. [[CrossRef](#)]
56. Parikh, S.; Dave, D.; Patel, R.; Doshi, N. Security and privacy issues in cloud, fog and edge computing. *Procedia Comput. Sci.* **2019**, *160*, 734–739. [[CrossRef](#)]