

Technologies Supporting Artificial Intelligence and Robotics Application Development

Yinong Chen^{1,2} and Gennaro De Luca³

¹School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, USA

²Faculty Ambassador, Amazon Web Services at ASU, Tempe, AZ 85287, USA

³Polytechnic School, Arizona State University, Mesa, AZ 85212, USA

(Received 09 December 2020; Revised 28 December 2020; Accepted 14 January 2021; Published online 31 January 2021)

Abstract: Artificial intelligence (AI) and robotics have gone through three generations of development, from Turing test, logic theory machine, to expert system and self-driving car. In the third-generation today, AI and robotics have collaboratively been used in many areas in our society, including industry, business, manufacture, research, and education. There are many challenging problems in developing AI and robotics applications. We launch this new *Journal of Artificial Intelligence and Technology* to facilitate the exchange of the latest research and practice in AI and technologies. In this inaugural issue, we first introduce a few key technologies and platforms supporting the third-generation AI and robotics application development based on stacks of technologies and platforms. We present examples of such development environments created by both industry and academia. We also selected eight papers in the related areas to celebrate the foundation of this journal.

Key words: artificial intelligence; robotics; machine learning; education

I. INTRODUCTION

Artificial intelligence (AI) and robotics research and application started in the 1950s. At that time, development and research of these two fields were still separate and independent [1]. The beginning of this first generation of AI is marked by Alan Turing's "Turing test," originally introduced as the "Imitation Game" by Turing in 1950. In his paper, Turing proposed a framework for machine intelligence and the test of that intelligence (i.e., the Turing test) [2]. However, the treatment of AI as a research area is often accredited to the 1956 Dartmouth conference, where McCarthy originally coined the term "Artificial Intelligence." At this time, Allen Newell, Cliff Shaw, and Herbert Simon were credited as creating one of the first AI programs, their logic theory machine [3]. Following this conference, AI grew in popularity, both as a result of the conference itself and the continual increase in computing power. While AI continued to grow in the computing sphere, robotics work was mainly in the realm of mechanical engineering with automated control systems, servos, and actuators. On the application side, AI was more in research while robotics was in manufacture. In the latter half of this generation, interest in AI began to dwindle due to factors such as insufficient computing power despite continual increases, insufficient data storage space and speed, and the resulting decrease in funding. This lull in research stalled AI for the last decade of the first generation.

The second generation started in the 1980s. Significant progress had been made in both AI and robotics. On the AI side, research interest was rekindled in part by the popularization of John Hopfield and David Rumelhart's "deep learning" techniques [4]. The formal introduction of expert systems by Edward Feigenbaum further added to this growth [5]. Feigenbaum's work on expert systems along with other advances in AI caused a significant increase in funding by the Japanese government who wanted to revolutionize computer processing and AI as part of their Fifth

Generation Computer Systems (FGCS) project. However, those goals were not met, resulting in the loss of the FGCS funding and a decrease in the popularity of AI [6]. Despite this loss of funding, the second generation of AI marks many successes for AI. These successes include the first self-driving car (Navlab) in 1986 by Carnegie Mellon, IBM's Deep Blue defeating chess grandmaster Gary Kasparov, and the implementation of Dragon Systems' speech recognition software into the Windows operating system (OS). Robotics similarly began to grow and support AI, such as by including sensors to add interactions with humans and the environment. Robotics applications have been extended from manufacturing alone to include support for businesses, services, and education.

The 2010s marks the third generation of AI, with AI and robotics merging in their theories, technologies, and applications. This third generation is marked by Moore's law approaching its end for classical computation, with quantum mechanics becoming an issue for further improvement [7]. Significant increases in processing and storage speed as compared to the previous generations of AI enable a new generation of both AI and robotics research and development. In this generation, AI and robotics are based on the following key technologies: big data-based intelligence, centralization and distribution optimization, neural networks and deep learning, from data collection to decision making in real time, hardware–software codefine, cognitive, natural language processing (NLP), human-machine interaction, and virtual and reality integration. The applications of AI and robotics have spanned across all domains of society, including healthcare, transportation, agriculture, business, finance, education, entertainment, defense, and government.

Both AI and robotics share the same subdomains, theories, technologies, and applications. Fig. 1 shows some of the domains in AI and robotics. All AI domains can be used in robotics. All domains in robotics, except the domains related to electronics, mechanics, and materials, are also applicable in AI.

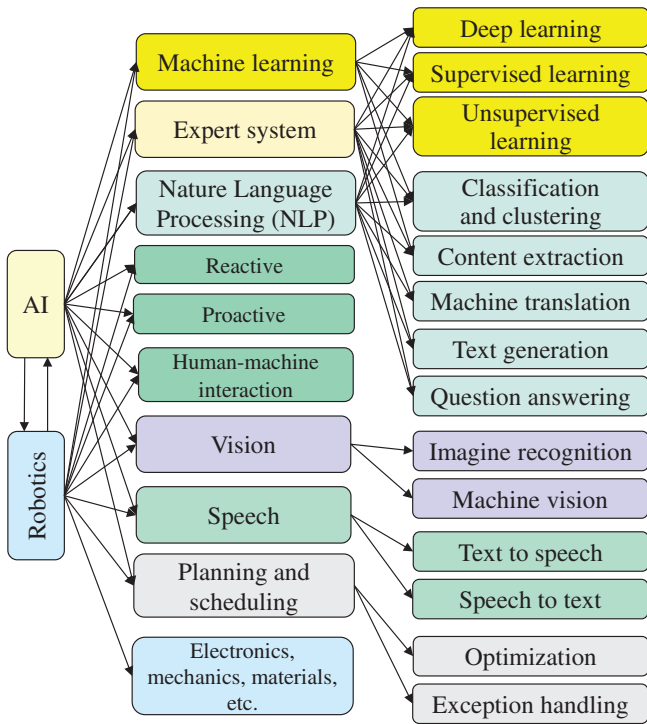


Fig. 1. Domains in artificial intelligence (AI) and robotics.

The third-generation AI and robotics are driven by tools, frameworks, and infrastructures. All the large IT corporations have created stacks of tools, frameworks, and infrastructures to facilitate the AI and robotics development and applications.

In the rest of the paper, we will introduce a few development stacks from major corporations in Section II. We will then discuss the AI and robotics development environment from Amazon Web Services (AWS) in more details. We will briefly present our research based on the AI and robotics development environment Visual IoT/Robotics Programming Language Environment (VIPLE) and AWS environment. Finally, we will present the papers selected for this inaugural issue of Artificial Intelligence and Technology.

II. AI AND ROBOTICS DEVELOPMENT ENVIRONMENTS

The third-generation AI and robotics are largely relying on powerful development environments. Many major IT corporations have created stacks of tools, frameworks, and infrastructures to facilitate AI and robotics development and applications. In this section, we present three AI development stacks from IBM, Nvidia, and Intel, respectively. They all have a strong hardware background and are leaders in AI software and hardware codesign.

A. IBM AI DEVELOPMENT STACK

IBM was a major computer hardware manufacturer since the 1950s, starting with mainframe computers, and then dominated the market with its IBM PCs. IBM exited the PC business and then the mainframe business in the early 2010s, and IBM converted itself into a service company focusing on offering business solutions and services. As AI becomes the power engine of new

Applications	Automotive and transportation, security and public safety, consumer web, mobile, and retail, medicine and biology, broadcast, media, and entertainment
Experience	NLP, knowledge representation and reasoning, deep learning, text to speech, and speech to text
Framework	Caffe, NVidia Caffe, IBM Caffe, Theano, Torch, TensorFlow, DL4J, and Chainer
Library and Tool	DIGITS, OpenBLAS, Distributed frameworks Bazel and NCCL
Hardware	IBM Power System for HPC, with NVLink GPC accelerated applications for deep learning SyNAPSE

Fig. 2. IBM AI development stack.

business solutions, IBM has offered a comprehensive solution with a stack of technologies as shown in Fig. 2.

At the hardware level, IBM uses NVidia NVLink for networking and reenters the hardware business to develop its own SyNAPSE hardware for vast neural networks for deep learning [8]. At the library and tool level, different AI libraries can be included in the IBM stack, including NVidia Deep Learning graphics processing unit (GPU) Training System, an open-source implementation of the Basic Linear Algebra Subprograms (OpenBLAS), Bazel, and NVidia Collective Communications Library.

At the framework level, different implementation of Convolutional Architecture for Fast Feature Embedding (Caffe) can be included. Caffe is a deep-learning framework, originally developed at the University of California, Berkeley, and is now implemented on different platforms. Developed by Google, TensorFlow is an open-source software library for machine learning, particularly focusing on training and inference of deep neural networks (DNN). Theano, Torch, DL4J, and Chainer are also widely used frameworks supporting machine learning and AI application development.

The IBM AI development stack offers rich experiences in NLP, knowledge representation and reasoning, deep learning, text to speech, and speech to text. Many applications have been developed based on the IBM AI stack, including automotive and transportation, security and public safety, consumer web, mobile, retail, medicine and biology, broadcast, media, and entertainment [9].

B. NVIDIA AI DEVELOPMENT STACK

As a leading GPU and AI solution provider, Nvidia offers powerful solutions on its own and also provides the other AI providers a range of hardware and software support. Fig. 3 shows the NVidia AI development stack [10].

At the hardware level, the Nvidia DGX SuperPO system consisting of Nvidia general-purpose computing on GPU-based workstations, servers, or cloud can be used as the infrastructure. The other hardware infrastructure from other vendors, such as Dell, HP, Lenovo, Cosco, IBM, AW, Alibaba, Google, Microsoft, etc., can also support Nvidia CUDA Toolkit running on their hardware infrastructures.

AI Frameworks	Cloud Services	Deployment	
Chainer, mxnet, PaddlePaddle, PyTorch, TensorFlow	accenture, AWS SageMaker, Google Cloud ML, Microsoft Azure Machine Learning, databricks	AWS SageMaker Neo, ONNG RUNTIME, TensorFlow Serving	
NVidia CUDA-X AI (Acceleration Libraries Speed Up Machine Learning)			
NVidia CUDA Toolkit			
NVidia DGX SuperPO, consisting of GPGPU-based Workstations, Servers, or Cloud	Workstations from other vendors: Dell, HP, Lenovo, etc.	Servers from other vendors: Cisco, DellEMC, HP Enterprise, IBM, Lenovo, etc.	Cloud from other vendors: AWS, Alibaba, Google Cloud, Microsoft Azure, etc.

Fig. 3. NVidia AI development stack.

NVIDIA CUDA is a parallel computing platform and programming model for general computing GPUs. With CUDA, developers can speed up computing applications by harnessing the power of GPUs.

The next level is the NVIDIA CUDA-X, which is built on CUDA. It is a collection of libraries, tools, and technologies that deliver dramatically higher performance than alternatives across multiple application domains—from AI to high-performance computing. A complete set of libraries and descriptions can be found in [11]. These libraries run everywhere from resource-constrained IoT and robotics devices, to self-driving cars, to the largest supercomputers and cloud computing environments.

On top of NVIDIA CUDA-X AI layer, different AI frameworks, cloud services, and deployment environments are supported to accelerate their performances. The AI frameworks include Chainer, mxnet, PaddlePaddle, PyTorch, and TensorFlow. The cloud services include many widely used services, including Accenture, AWS SageMaker, Google Cloud ML, Microsoft Azure Machine Learning, and Databricks. The deployment environments include AWS SageMaker Neo, ONNG RUNTIME, and TensorFlow Serving.

C. INTEL AI DEVELOPMENT STACK

As a leading hardware provider, Intel has transformed into a software, AI, robotics, and AI solution provider. Fig. 4 shows the Intel AI development stack [12].

At the hardware level, Intel offers AI enabled processor, memory and storage, field-programmable gate array, grade point average (GPA), network solution, and real-sense devices.

At the library level, Intel developed a variety of libraries that specifically use their AI hardware features to accelerate AI computing, including DAAL, Nervana graphic compiler, MKL-DNN, MLSL, Movdius, and MvTensor libraries.

At the framework level, common AI frameworks are supported, including BigDL: Distributed Deep Learning on Apache Spark, TensorFlow, Caffe, theano, and torch.

At the tool level, Intel developed tools for supporting AI and robotics application development, including Intel Deep Learning software development kit (SDK), Computer Vision SDK, Movdius, and saffron.

At the experience level, Intel demonstrated variable applications based on this stack of technologies, including Intel autonomous car, drone, robot, game, finance, medical, and security systems.

Experience	Intel autonomous car, drone, robot, game, finance, medical, and security systems
Tool	Intel Deep Learning SDK, Computer vision SDK, Movdius, and saffron
Framework	BigDL: Distributed Deep Learning on Apache Spark, TensorFlow, Caffe, theano, and torch
Library	DAAL, Nervana graphic compiler, MKL-DNN, MLSL, Movdius, and MvTensor libraries
Hardware	AI Processor, memory & storage, FPGA, GPA, AI network solution, real sense devices

Fig. 4. Intel AI development stack.

III. AWS ROBOMAKER FOR EDUCATION

In the previous section, we presented the IBM, Nvidia, and Intel AI development stacks. They all have a strong hardware background and are leaders in AI software and hardware codesign. In this section, we will introduce AWS RoboMaker education platform and its solutions to AI and robotics development and applications [13].

AWS's strengths are in cloud computing, big data processing, as well as web and AI services. It does not have extensive hardware solution experiences. The AWS solution uses Robot Operating System (ROS), the Gazebo physics engine, and NVidia GPU-based robots to build its OS, simulation, and physical robot environments.

ROS is the most widely used software framework for teaching and learning about robotics. It had over 16 million downloads in 2018, a 400% increase since 2014. ROS is primarily a set of software libraries and tools, from drivers to algorithms, that help developers build robot applications.

From the robotics education perspective, ROS is used by students of all ages, from kids interacting with robots in museum exhibits to graduate students learning about the latest solutions to common robotics problems. Because it supports such a wide variety of robots, including low-cost platforms like the TurtleBot and LEGO Mindstorms, ROS is especially well suited for classroom use.

Gazebo is a robust physics engine with the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It offers high-quality graphics and convenient programmatic and graphical interfaces to help developers simulate robots.

Although ROS and Gazebo are excellent robotics development environments, they do not offer much AI functionality and support. The main contribution from AWS in the RoboMaker environment is the integration of AWS resources into the robotics development environments, including the cloud extensions and development tools. The integrated resources offer the benefits of accelerated development timelines, intelligence out of the box, zero infrastructure provisioning, and lifecycle management of intelligent robotics applications.

AWS cloud extensions are written as ROS packages that automatically create connections and make application programming interface (API) calls to AWS services, such as

- AMAZON LEX speech recognition
- AMAZON POLLY speech generation

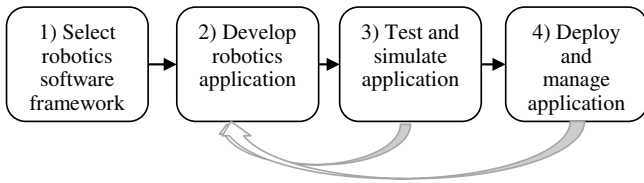


Fig. 5. Robotics application development iterations.

- KINESIS VIDEO STREAMS video streams
- AMAZON REKOGNITION image and video analysis
- CLOUDWATCH logging and monitoring

Developing intelligent robotics applications is time consuming and challenging. It requires machine-learning expertise for intelligent functions. It requires many prototyping iterations. It can take days to set up and configure the software and hardware. It can take months to build a realistic simulation environment for testing the applications. Fig. 5 shows a basic development cycle of robotics applications, in which Steps 2, 3, and 4 require many iterations.

AWS RoboMaker and its artificial intelligence services make it easy for developers to develop, test, and deploy robotics applications. The RoboMaker environment is designed for every level of developer and learner, including application developers, quality engineers, system engineers, and particularly for researchers and students.

To support AI and robotics education, AWS offers free-tier accounts that offer adequate resources for students to use AWS resources [14], for example, 750 hr per month, one million accesses to AWS Lambda IoT API calls, 25 GB Amazon Dynamo DB storage, 3 GB Amazon S3 storage, etc.

For instructors using AWS services for teaching, teacher accounts can be created. A teacher can request additional credits for students in the class to use resources beyond the free-tier accounts.

AWS offers a variety of OSs and platforms for developers and students to use in its cloud-computing environments. Fig. 6 shows examples of the available OSs and platforms.

IV. ARIZONA STATE UNIVERSITY (ASU) IoT/ROBOTICS DEVELOPMENT ENVIRONMENT

ASU has developed a VIPLE that offers an array of general-purpose computing, IoT/Robotics programming, service-oriented computing, and AI-based computing [15]–[17]. VIPLE can be used with different simulators and physical robots, including maze simulator, traffic simulator, race car simulator, and different physical robots based on Intel, Raspberry Pi, and Lego robots. Fig. 10 shows a few of the simulators and physical robots integrated into VIPLE.

A. GENERAL PURPOSE AND SERVICE COMPUTING IN VIPLE

VIPLE supports general-purpose programming through a set of basic activities, as shown in Fig. 6, which include data, variable, calculate, while, and switch. It also includes merge and join for

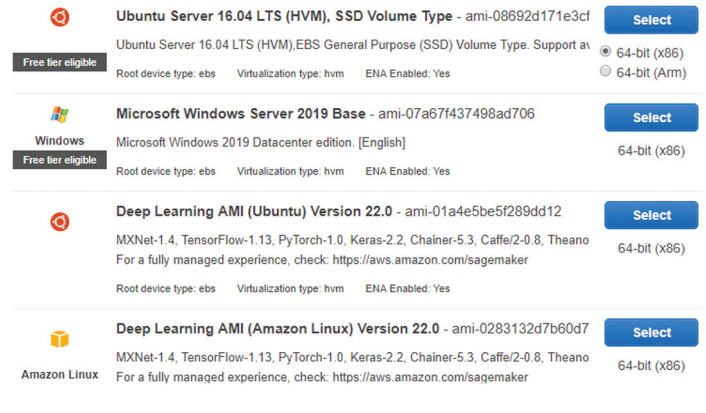


Fig. 6. Operating System (OS) and frameworks available in Amazon Web Services (AWS) cloud.

performing parallel computing. The services list at the bottom of Fig. 7 will be further explained in the following sections.

A program in VIPLE is constructed visually through drag and drop from the Basic Activities list and from the other service lists. Fig. 8 shows the implementation of a program that add 1500 numbers in a while loop. In this program, two variables, sum of type Double and *i* of type Int32, are used. Both variables are initialized to 0. Then, in a while loop, *i* is added to sum from *i* = 1 to *i* = 1500. The result is printed after exiting the loop.

VIPLE is extensible. Different components, in the forms of code activities, VIPLE services, and web services, can be added into the VIPLE. It can support AI-based computing through Web Services Description Language and RESTful services. Fig. 8 shows the addition of image recognition services into the VIPLE service list. Once the service is added, we can feed an image URL or an



Fig. 7. Visual IoT/Robotics Programming Language Environment (VIPLE) basic activities for general-purpose computing.

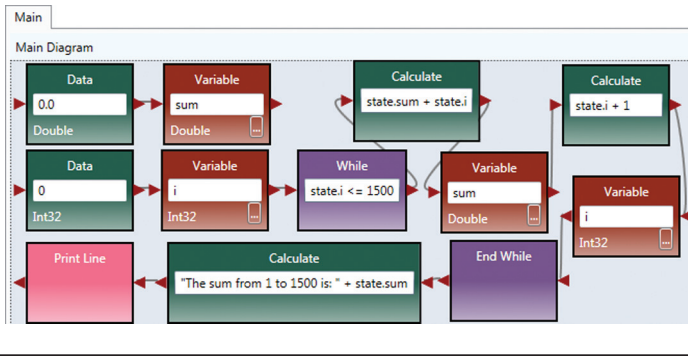


Fig. 8. VIPLE performing general-purpose computing.

image stream to the service, which will then recognize the image. In the example, an image of a stop sign and an image of a dog are input to the service, and the recognition results are generated with certainty probabilities.

In the training of our image recognition services, we trained by shapes without text recognition, and thus, the stop sign shape is common in different things, such as boxes of face powder or perfume, or a book jacket, the recognition probability is split. For the dog, the recognition is accurate. It not only recognizes a dog, but also recognizes the breed of the dog.

From the general-purpose services list on the left side in Fig. 9, we can see the other type of components available for customizing the general-purpose services list, including Code Activity C# and Code Activity Python. We will discuss using code activity in the next subsection for programming an autonomous driving race car.

B. IoT/ROBOTICS PROGRAMMING IN VIPLE

As discussed in the previous section, AWS RoboMaker is intended to make robotics application development easier and faster. VIPLE is designed for the same purpose. Since VIPLE uses the visual programming paradigm, it is even easier than AWS RoboMaker.

VIPLE supports a number of IoT and robotics simulation and physical environments. Fig. 10 shows a few examples supported by VIPLE. At the bottom layer of the figure are three maze simulation environments. The first two are implemented in HTML5 as 2D and 3D web applications. The third simulator is implemented in the Unity 3D programming environment. The next layer above are the traffic simulator and a race car simulator. The traffic simulator is also implemented in the Unity 3D programming environment. The race car simulator is the open-source project The Open Racing Car Simulator and is a highly portable multiplatform car-racing simulation. It can be used as an ordinary car-racing game, an AI racing game, and as a research platform [18].

The layers above are the hardware templates. VIPLE can directly control Lego EV3 robots through a Bluetooth or Wi-Fi connection. We also implemented open architecture robots and drones using Intel Galileo and Edison, as well as Raspberry Pi. We also used commercial-off-the-shelf BIOLOID robots that can be configured into a humanoid or a robot dog.

The VIPLE visual programming environment is easier to use but it does not directly offer many AI-based services and functions. The solution is to use web services to access the external services. We have been working with AWS to use its AI services to extend VIPLE's capacity.

A flight path machine-learning application was developed using the drone in Fig. 10, which uses VIPLE and AWS AI services [19].

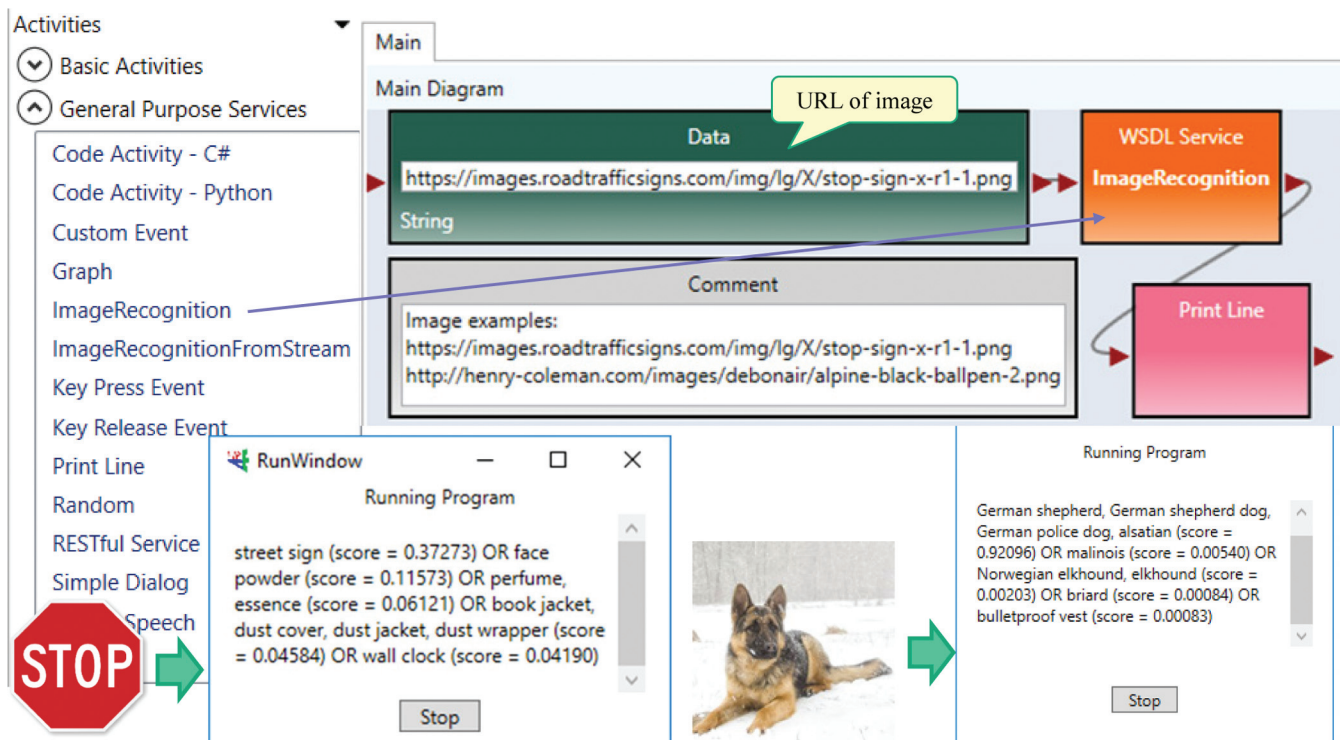


Fig. 9. Image recognition service in VIPLE.

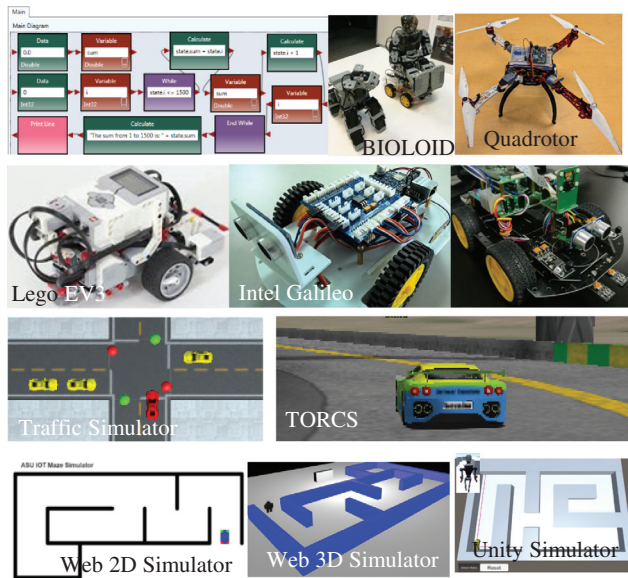


Fig. 10. VIPLE-supported simulators and physical robots.

A guide dog application for vision impaired was developed using the BIOLOID robot in Fig. 10, which uses VIPLE and AWS Alexa voice services and AI services [20].

V. PAPERS IN THIS INAUGURAL ISSUE

There are many challenging problems in developing AI and robotics applications. We launch this new *Journal of Artificial Intelligence and Technology* to facilitate the exchange of the latest research and practice in AI and technologies. As we discussed in the previous sections, the third-generation AI is based on stacks of technologies and platforms.

In this inaugural issue, we first introduced a few key technologies and platforms supporting AI and robotics application development. We also selected a few papers in the related areas to celebrate the foundation of this journal. These papers fall into two areas: AI and robotics-based education systems and image processing and recognition in AI and robotics system.

A. AI AND ROBOTICS-BASED EDUCATION SYSTEMS

We selected two papers in this topic area that help improve teaching efficiency.

The first paper attempts to address class attendance through AI technology [21]. Class attendance registering is often done using “roll-call” or signing sheet. These methods are time consuming, easy to cheat, and it is difficult to draw any information from it. There are other, expensive alternatives to automate attendance registering with varying accuracy. This study experimented with a smartphone camera and different combinations of face detection and recognition algorithms to determine if it can be used to successfully mark attendance while keeping the solution cost effective. The effect of different class sizes was also investigated. The research was done within a pragmatism philosophy, using a prototype in a field experiment. The algorithms that were used: Viola-Jones (HAAR features), DNN, and histogram of oriented gradients for detection and Eigenfaces, Fisherfaces and Local

Binary Pattern Histogram (LBPH) for recognition. The best combination was Viola-Jones together with Fisherfaces with a mean accuracy of 54% for a class of 10 students and 34.5% for a class of 22 students. The best all-over performance on a single class photo was 70% (class size 10). As is, this prototype is not accurate enough to use, but with a few adjustments, it may become a cheap, easy to implement solution to the attendance marking problem.

Teaching students the concepts behind computational thinking is a difficult task, often gated by the inherent difficulty of programming languages. In the classroom, teaching assistants may be required to interact with students to help them learn the material. Time spent grading and offering feedback on assignments are removed and used to help students directly. The second paper selected attempts to address the problem by offering a framework for developing an explainable AI that performs automated analysis of student code while offering feedback and partial credit. The creation of this system is dependent on three core components. Those components are a knowledge base, a set of conditions to be analyzed, and a formal set of inference rules. In this paper, such a system is developed based on a VIPLE language [16], [17], Pi-Calculus, and Hoare Logic. The detailed system can also perform self-learning of rules. Given solution files, the system can extract the important aspects of the program and develop feedback that explicitly details the errors students make when they veer away from these aspects. The level of detail and expected precision can be easily modified through parameter tuning and variety in sample solutions [22].

B. PLANNING AND DECISION

Two papers are selected in this topic area that present theoretical studies to support planning and decision making in AI applications.

The first paper is on the continuum robot manipulators that have great potential applications in the fields of minimally invasive surgeries, home services, endoscope detections, field rescues, etc. The kinematics, dynamics, and control issues of the continuum manipulators are rather different from a conventional rigid-link manipulator. By the aid of Lie groups theory and exponential coordinate representations, the kinematics of the continuum manipulators with piecewise constant curvatures and actuated by tendons is investigated in this paper. Based on differential kinematics analyses, the complete Jacobian of the continuum manipulators is derived analytically, and then a motion planning approach named as “Dynamic Coordination Method” is presented for the continuum manipulators. The feasibility of the modeling and the motion planning method are demonstrated by some numerical simulations [23].

The second paper studies aggregation operators that are useful in converting all individual input data into a single argument and have a great importance in applications involving decision making, pattern recognition, medical diagnosis, and data mining. This paper presents a new concept named as Cubic q-rung orthopair fuzzy linguistic set to quantify the uncertainty in the information. The proposed method is qualitative form of cubic q-rung orthopair fuzzy set, where membership degrees and nonmembership degrees are represented in terms of linguistic variables. The basic notions of proposed method have been introduced and their basic operations and properties studied. Furthermore, the paper aggregates the different pairs of the preferences. It introduces the Muirhead mean, weighted Muirhead mean, and dual Muirhead mean-based operators. The major advantage of considering the Muirhead mean is that it considers the interrelationship between more than two arguments at a time. The method also has the ability to describe the

qualitative information in terms of linguistic variables. Several properties and relation of the derived operators are argued. In addition, the paper also investigates multiattribute decision-making problems under the developed environment and illustrates with numerical examples. Finally, the effectiveness and advantages of the work are established by comparing with other methods [24].

C. AI AND ROBOTICS-BASED OBJECT DETECTION

Two papers are selected in this topic area that deal with object detection and recognition based on image processing and machine learning.

The first paper is on human activity recognition and embedded application based on convolutional neural network. With the improvement of people's living standards, the demand for health monitoring and motion detection is continuously increasing. It is of great significance to study human activity recognition methods, which is different from the traditional feature extraction methods in many ways. This paper uses the convolutional neural network algorithm in deep learning to automatically extract the characteristics of human's daily activities and uses the stochastic gradient descent algorithm to optimize the parameters of the convolutional neural network. The network model is compressed on existing methods. The paper presents the recognition of six kinds of human daily activities, such as walking, sitting, standing, jogging, going upstairs, and going downstairs using neural networks on embedded devices [25].

The second paper selected in this topic area is on obstacle detection method of urban rail transit based on multisensor technology. With the rapid development of urban rail transit, passenger traffic keeps increasing, and obstacle violations are more and more frequent. The safety of train operation under high-density traffic conditions is becoming more serious than ever. To monitor the train operating environment in real time, this paper adopts a multi-sensing technology based on machine vision and lidar, which are used to collect video images and ranging data of the track area in real time. Then, image preprocessing and division of regions of interest on the collected video are processed. The obstacles in the region of interest are detected to obtain the geometric characteristics and position information of the obstacles. Finally, according to the danger level of the obstacles, the degree of impact on train operation is determined, and the automatic response mode and manual response mode of the signal system are used to transmit the detection results to the corresponding train to control train operation. Through simulation analysis and experimental verification, the detection accuracy and control performance of the detection method are confirmed, which provide safety guidance for the train operation [26].

D. IMAGE PROCESSING AND OBJECT RECOGNITION

Two papers are selected in this topic area that deal with object detection and recognition based on image processing and machine learning.

The first paper selected in this topic area is on multiscale center point object detection based on parallel network. Anchor-based detectors are widely used in object detection. To improve the accuracy of object detection, a large number of anchor boxes are intensively placed on the input image. However, most anchor boxes are invalid. Although the anchor-free method can reduce the

number of useless anchor boxes, the invalid boxes still occupy a high proportion. Based on this observation, this paper proposes a multiscale center point object detection method making use of parallel network, which can further reduce the number of useless anchor boxes using anchor-free mode. This paper adopts the parallel network architecture of hourglass-104 and darknet-53. Hourglass-104 outputs heat map to generate the center point for object feature location on the output attribute feature map of darknet-53. The algorithm combines the feature pyramid and the complete intersection-over-union (CIOU) loss function. The algorithm is trained and tested on Microsoft common objects in context (MSCOCO) dataset, which can increase the detection rate of target location and improve the accuracy rate of small object detection. Its overall object detection accuracy rate is similar to the state-of-the-art two-stage detectors method, but it has the advantage of speed [27].

The second paper presents a fully convolutional neural network-based regression approach for effective chemical composition analysis using near-infrared spectroscopy (NIR) in cloud-computing environment. It uses a one-dimensional fully convolutional network model to quantitatively analyze the nicotine composition of tobacco leaves using NIR data in a cloud environment. The one-dimensional convolution layers directly extract the complex features from sequential spectroscopy data. It consists of five convolutional layers and two full connection layers with the max-pooling layer replaced by a convolutional layer to avoid information loss. Cloud-computing techniques are used to solve the increasing requests of large-size data analysis and implement data sharing and accessing. Experimental results show that the proposed one-dimensional fully convolutional network model can effectively extract the complex characteristics inside the spectrum and can more accurately predict the nicotine volumes in tobacco leaves than the existing approaches. This research provides a deep-learning foundation for quantitative analysis of NIR spectra data in the tobacco industry [28].

VI. CONCLUSION

There are many challenging problems in the third-generation AI and robotics applications. In this editorial paper, we introduced the development of AI and robotics theories and technologies, particularly the current AI and robotics development environments based on stack of technologies, including those from IBM, Intel, and Nvidia. We presented AWS RoboMaker and ASU VIPLE for developing integrated AI and robotics solutions. We also selected eight papers in the related areas for this inaugural issue of this journal.

REFERENCES

- [1] Y. Chen, *Service-Oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services*, 7th ed., Dubuque, IA, USA: Kendall Hunt Publishing, 2020.
- [2] A. M. Turing, "Computing machinery and intelligence," *Parsing the Turing Test*, R. Epstein, G. Roberts, and G. Beber, Eds. Springer, Dordrecht, 2009, pp. 23–65.
- [3] J. Moor, "The Dartmouth College artificial intelligence conference: The next fifty years," *AI Magazine*, vol. 27, no. 4, pp. 87–87, 2006.
- [4] K. Gurney, *An Introduction to Neural Networks*, Boca Raton, Florida, USA: CRC Press, 1997.
- [5] E. A. Feigenbaum, "Some challenges and grand challenges for computational intelligence," *J. ACM*, vol. 50, no. 1, pp. 32–40, 2003.

- [6] E. A. Feigenbaum and P. McCorduck, *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World*. Reading, MA: Addison-Wesley Pub. Co., 1983.
- [7] J. L. Hennessy and D. A. Patterson. "A new golden age for computer architecture," *Commun. ACM*, vol. 62, no. 2, pp. 48–60, 2019.
- [8] IBM, New IBM SyNAPSE Chip Could Open Era of Vast Neural Networks, Available: <https://www-03.ibm.com/press/us/en/pressrelease/44529.wss>
- [9] K. Lui and J. Karmiol, "AI Infrastructure Reference Architecture IBM Systems," 87016787USEN-00, IBM, Jun., 2018, Available: <https://www.ibm.com/downloads/cas/W1JQBNJV>
- [10] Nvidia CUDA-X, Available: <https://www.nvidia.com/en-us/technologies/cuda-x/>. Accessed on: November 2020.
- [11] Nvidia CUDA-X libraries, Available: <https://developer.nvidia.com/gpu-accelerated-libraries>. Accessed on: November 2020.
- [12] Intel, Simplify AI with Intel® Artificial Intelligence Technologies, Available: <https://www.intel.com/content/www/us/en/artificial-intelligence/overview.html>
- [13] AWS RoboMaker, Available: <https://aws.amazon.com/robomaker/>
- [14] AWS Free Tier, Available: <https://aws.amazon.com/free/>
- [15] VIPLE, Available: <http://neptune.fulton.ad.asu.edu/VIPLE/>
- [16] Y. Chen and G. De Luca, "VIPLE: Visual IoT/Robotics Programming Language Environment for Computer Science Education", in *IPDPS Workshops*, 2016, pp. 963–971.
- [17] Y. Chen and G. De Luca, "Technologies for developing a smart city in computational thinking," *Int. J. Sim. Process Model*, vol. 13, no. 2, 2018, pp. 91–101.
- [18] TORCS—The Open Racing Car Simulator, Available: <https://sourceforge.net/projects/torcs/>. Accessed on: November 2020.
- [19] M. De La Rosa and Y. Chen, "A Machine Learning Platform for Multirotor Activity Training and Recognition," in *Proc. of IEEE 14th ISADS*, Utrecht, the Netherlands, Apr. 2019, pp. 15–22.
- [20] J. Zhu, J. Hu, M. Zhang, Y. Chen, and S. Bi "A fog computing model for implementing motion guide to visually impaired," *Simul. Model. Pract. Theor.*, vol. 101, May 2020.
- [21] L. Cronjé and I. Sanders, "Semi-automated class attendance monitoring using smart phone technology," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021, DOI: 10.37965/jait.2020.0040.
- [22] G. De Luca and Y. Chen, "Explainable AI for workflow verification in VIPLE," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0023.
- [23] J. Zhang, J. Yuan, J. Di, and G. He, "Kinematics modeling and motion planning of tendon driven continuum manipulators," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0041.
- [24] H. Garg, T. Mahmood, J. Ahmmad, Q. Khan, and Z. Ali, "Cubic q-rung orthopair fuzzy linguistic set and their application to multi-attribute decision making with Muirhead mean operator," *Intell. Sci. Technol. Press*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0062.
- [25] Y. Xu and T. Qiu, "Human activity recognition and embedded application based on convolutional neural network," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0051.
- [26] T. Xiao and Y. Xu, "Research on obstacle detection method of urban rail transit based on multi-sensor technology," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0027.
- [27] H. Chen, H. Zheng, and X. L. Li, "Multiscale center point object detection based on parallel network," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0025.
- [28] D. Jiang, G. Hu, G. Qi, and N. Mazur, "A fully convolutional neural network-based regression approach for effective chemical composition analysis using near-infrared spectroscopy in cloud," *J. Artif. Intell. Technol.*, vol. 1, no. 1, Jan. 2021. DOI: 10.37965/jait.2020.0037.



YINONG CHEN is currently a principal lecturer in the School of Computing, Informatics, and Decision Systems Engineering at Arizona State University. He received his doctorate from the University of Karlsruhe/Karlsruhe Institute of Technology (KIT), Germany, in 1993. He did postdoctoral research at Karlsruhe and at LAAS-CNRS in France in 1994 and 1995. From 1994 to 2000, he was a lecturer and then

senior lecturer in the School of Computer Science at the University of the Witwatersrand, Johannesburg, South Africa. Chen joined Arizona State University in 2001. He's (co-) authored 10 textbooks and more than 200 research papers. He is the editor-in-chief of the *Journal of Artificial Intelligence and Technology* and on the editorial boards of several journals, including *Journal of Systems and Software*, *Simulation Modeling Practice and Theory*, and *International Journal of Simulation and Process Modelling*. Chen's areas of expertise include software engineering, service-oriented computing, robotics and artificial intelligence (AI), visual programming, dependable computing, and computer science education.



GENNARO DE LUCA is currently a lecturer in the Polytechnic School at Arizona State University. He received his Ph.D. from Arizona State University in May 2020. His research areas include visual programming, artificial intelligence systems, reasoning, and process calculi. His doctoral research focused on the application of AI reasoning techniques for semantic analysis of workflow programs. He was the first-prize winner

in July 2016 Intel Cup in Shanghai, a finalist in Intel Cornell Cup, May 2016 in Orlando, and a finalist in Microsoft Imagine Cup March 2016 in San Francisco.