Volume 4, Issue 2, August 01, 2019

TC-CPS Newsletter

IEEE TC-CPS Cyber-Physical Systems

An Introduction to Trajectory Tracking Control for Autonomous Underwater Vehicle

Heng Zhang, Weiwei Xu, Jian Zhang, Hongran Li, Yunling Li,
Jiangsu Ocean University

1 Introduction

Autonomous underwater vehicle (AUV), as a cable-free underwater vehicle, is an important tool for the exploration and development of marine resources. It applies artificial intelligence, automatic control, pattern recognition, information fusion and system integration and other technologies to traditional carriers [1]. It does not have physical connection with the mother ship. It can complete the scheduled tasks with its own power and machine intelligence under the consideration of unmanned circumstance.

AUV is widely applied in the field of military, including anti-submarine warfare, mine warfare, intelligence reconnaissance, patrol surveillance, logistics support, topographic mapping and underwater construction [2, 3, 4]. Therefore, all countries are committed to the research of advanced AUV system to enhance national defense capability. In America, the Navy Space and Naval Warfare Systems Center, the Navy Research Institute, the Massachusetts Institute of Technology Marine and many other research institutions have developed a large number of AUVs which are applied to short term and long term mine reconnaissance system. In 1990, the Norwegian Defense Research Institute made a long-term development plan for the AUV, among which the HUGlN series AUVs have participated in the mine-hunting demonstration of the Royal Norwegian Navy for many times. Meanwhile, Britain, France, Germany, Russia and other countries have also carried out a lot of research work on the AUV in military [5].

In the civil field, AUV is mainly used for marine environment investigation, exploration of seabed mineral and biological resources, maritime rescue, marine archaeology and construction and maintenance of submarine optical cable project [6]. With the support of China Ocean Mineral Resources Research Association, China and Russia jointly developed the CR-01 and CR-02 AUV, which can complete acoustic, optical and hydrological surveying tasks in the polymetallic working environment of the Submarine flat terrain. In 2013, the Tesla offshore has used the "bluefin-21" AUV to carry out the offshore pipeline maintenance services and the mapping of seabed geographical environment. In 2014, the "bluefin-21" AUV played an important role in the search for the missing Malaysia airlines flight 370.

In 2015, the "Dorado" AUV was launched in the Juan DE Fuca Ridge by the American Monterey Marine Research Institute, which has a minimum sailing height of 50m above the seabed. The AUV is equipped with multi-beam sonar that can accurately map the seabed topography around the crater. In 2016, the "VideoRav" AUV developed by Lnuktun of Canada successfully completed the underwater biological monitoring of the inlet of the marine cooling system in the Daya Bay nuclear power plant. In the case of strong flow and complex underwater conditions, the AUV successfully obtained the pictures of the inlet of the cooling system by relying on its excellent underwater resistance performance and strong underwater power.

To ensure the successful completion of underwater environmental survey, underwater search, underwater rounding up, underwater transmission and other control tasks, the tracking control problem is needed to solved firstly. The tracking control problem is an important aspect of underwater vehicle technology, it mainly includes the path following and the trajectory tracking. The main difference of which is that a tracking trajectory is related to time or not. The former has nothing to do with time and the latter is related to time. The path following can be viewed as a special case of the trajectory tracking. Due to the difficulty of accurately obtaining the self-dynamic model and the

characteristics of strong coupling and nonlinearity, conventional model-based control strategies of AUV are difficult to meet the control requirements. In addition, compared with the condition of ground mobile vehicles, the complex underwater environment and time-varying ocean currents also make the research of underwater trajectory tracking and control more challenging.

2 The trajectory tracking technology of AUV

In view of the complex and changeable marine environment, the main research methods of trajectory tracking and control of AUV at present include PID control, adaptive control, sliding mode control, neural network control, backstepping control, etc.

2.1 PID control

PID control is a control strategy for single input single output system which was established based on the linear system theory. Its controller design is relatively simple, while the system of AUV is belong to multiple input and multiple output nonlinear system, and the complex environment time-varying interference is also a problem when the AUV works at the bottom of the ocean. Therefore, single PID control mode in the motion control of the AUV has been unable to obtain ideal control effect [7, 8]. Usually the PID algorithm was combined with other intelligent algorithms, such as PID neural network controller proposed by Huang et. al. [9]. A beneficial motion control which is based on two degree-of-freedom PID and extreme learning machine for AUV was proposed by Liu et. al. [10], and a fractional order PID strategy that is based on seeker optimization algorithm for AUV was proposed by Wan et. al. [11].

2.2 Adaptive control

Adaptive control can improve the robustness of controlled system. Compared with general controller, the parameters of adaptive controller are changing, and it has a mechanism which can automatically on-line correct these parameters according to the signal of system. When the parameters change, the controller can adjust the control rules of system by learning and identification on time [12]. At present, the application of adaptive control in nonlinear objects is relatively limited. In most cases, more applications have been obtained in the control of AUV by combining adaptive control with other control methods. Wang et. al. presented an Self-adaptive path following control which was based on virtual guidance [13]. Li et. al. put forward to a modified adaptive hybrid fuzzy control algorithm with Mamdani inference [14]. Hu et. al. proposed a command filtering based adaptive fuzzy backstepping method [15]. Jiang et. al. designed an adaptive control strategy which can solve the horizontal trajectory tracking problem in consideration of parameters perturbation and current disturbances [16]. However, its structure is more complicated, and it is difficult to obtain a unified standardization method for designing controller.

2.3 Sliding mode control

The basis of the sliding mode control is to design a proper switching function and control law, which can make the system state trajectory to reach the designed switching manifold under the limited time and slide to balance with the appropriate speed [17]. After entering the ideal sliding mode, the system have a stronger robustness to external disturbance. Due to its robust control performance, sliding mode control is insensitive to the changeable parameters and can suppress disturbances. It does not require accurate modeling of dynamic model, so it is often used for dynamic tracking control of AUV. Jiang et. al. utilized integral sliding mode control to deal with the problem of horizontal trajectory tracking under the condition of parameter perturbation [18]. Zhang et. al. presented a terminal sliding mode variable structure control system which was used to solve the nonlinear control problem of AUV with changeable parameters [19]. Konar et. al. designed a fractional order sliding mode controller for depth control of AUV [20]. However, the critical problem of sliding mode control is its high frequency switching control behavior (buffeting problem). Buffeting problem makes heat loss highly in the electrical power circuit and motor, which affects the accuracy of underwater vehicles.

2.4 Neural network control

Neural network control is a control strategy which is produced by simulating the function and structure of the nerves in human brain. It can fully approach any complex nonlinear system and use neural network to fit the nonlinear performance of AUV. Neural network control has the characteristics of nonlinear, self-learning and other intelligent characteristics, which can adapt to and learn the dynamic characteristics of the system through adjusting the weight. This feature is very suitable for the motion control of AUV. For instance, an adaptive neural network control of AUV which controlled input nonlinearities through using reinforcement learning was investigated by Cui et. al. [21]. Miao et. al. proposed that the error could be calculated by applying radial basic function neural networks. They constructed an adaptive neural network controller by minimal learning parameter and dynamic surface control. This proposed method could track the stable trajectories [22]. However, neural networks control is not only difficult to obtain training samples, but also lags in the learning process of samples, which make the real-time performance of the control system poor.

2.5 Backstepping control

Backstepping controller is widely used in the tracking control of mobile vehicle. Now, it has been applied to the control system of AUV. The backstepping algorithm can stabilize the closed-loop control system by designing the speed controller under the condition of large initial error. Its design is relatively simple and can be rigorously proved by lyapunov stability theory. For instance, Zhang et. al. proposed a backstepping control approach to realize the 3D trajectory tracking under the condition of external interference [23]. Gharesi et. al. designed a backstepping controller to track the trajectory in the desired depth [24]. Cervantes et. al. put forward to a controller which output is based on backstepping control to tracking the trajectories [25]. Xu et. al. proposed a intelligent system of underwater salvage where the controller of AUV was based on backstepping control [26]. However, due to the design of the backstepping control is directly related to the state error, it will be generated larger changeable speed under the larger initial state error. In another word, the AUV usually face the phenomenon of jumping speed when the motion state have a jumping change. Considering the dynamic factors, the required acceleration and force in jumping points may be out of control constraints when the backstepping control law is applied.

3 Future research

The study of AUV has far-reaching implications for enhancing the capability of detecting marine resources [27, 28]. In the future, after the technology of trajectory tracking for single AUV is mature, it should be considered to improve the tracking performance through researching the formation control of multi-AUVs. We will obtain the whole tracking performance by dynamic researching the adaptive formation control of multi-AUVs. However, the research for multi-AUVs has the following difficulties, which is also the direction of future research.

- The limitations of underwater communication and the problem of asynchronous transmission between between the AUVs in formation.
- The difficulties on the distributed control of multi-AUVs.
- The study on the adaptive control of formation network for multi-AUVs.
- The research on path planning of AUVs under optimal control.

In the process of navigation, the off-line global route can be used as a reference trajectory. The horizontal and vertical tracking can be applied respectively to establish an objective function for online tracking. The optimal control of the tracking can be realized through constructing the Hamilton function to calculate the tracking error in which the linearized lateral motion equation of AUV is acted as constraint. At the same time, we can solve the problem by employing the differential equations of classical variational extremum which can combine the methods of gradient iteration and one-dimensional search.

References

[1] W. Xu, Y. Xiao, H. Li, J. Zhang, H. Zhang, Trajectory tracking for autonomous underwater vehicle based on model-free predictive control, in *Procedings of IEEE 20th International Conference on High Performance Switching and Routing (HPSR)*, 2019.

[2] Y. Xu, Y. Su, Y. Pang, Expectation of the development in the technology on ocean space intelligent unmanned vehicles, *Chinese Journal of Ship Research*, 1(3):1–4, 2006.

[3] C. Simpkins, Introduction to autonomous manipulation£ºcase study with an underwater robot, *IEEE Robotics and Automation Magazine*, 21(4): 109–110, 2014.

[4] D. Li, Applications of wavelet algorithm in ship motion control system, *Ship Science And Technology*, 38: 49–51, 2016.

[5] A. Diercks, M. Woolsey, R. Jarnagin, V. Asper, C. Dike, M. Emidio, S. Tidwell, A. Conti, Site reconnaissance surveys for oil spill research using deep-sea AUVs, *Oceans-San Diego*, 1–5, 2013.

[6] R. Wernli, AUV commercialization¡ªwho¡¯s leading the pack, *IEEE Conference and Exhibition*, 1:1–4, 2000.

[7] G. Antonelli, Adaptive/integral actions for 6-DOF control of AUVs, in *Proceedings of IEEE International Conference on Robotics and Automation*, 3214–3219, 2006.

[8] Z. Tang, Q. He, S. Wang, An improved generalized predictive control for AUV yaw, in *Proceedings of International Conference on Mechatronics and Intelligent Materials*, 1709–1713, 2012.

[9] R. Huang, N. Ding, AUV vertical plane control based on improved PID neural network algorithm, *Journal of System Simulation*, 2019.

[10] W. Liu, X. Ding, J. Wan, R. Nian, B. He, Y. Shen, T. Yan, An effective motion control based on 2-DOF PID and ELM for AUV, *OCEANS 2018 MTS/IEEE Charleston*, 1–4, 2018.

[11] J. Wan, W. Liu, X. Ding, B. He, R. Nian, Y. Shen, T. Yan, Fractional order PID motion control based on seeker optimization algorithm for AUV, *OCEANS 2018 MTS/IEEE Charleston*, 26–29, 2018.

[12] G. Antonelli, S. Chiaverini, N. Sarkar, Adaptive control of an autonomous underwater vehicle: Experimental results on ODIN, *IEEE Transactions on Control Systems Technology*, 9(5): 756-765, 2001.

[13] H. Wang, Z. Chen, X. Bian, H. Jia, G. Xu, Robust adaptive path following control for autonomous underwater vehicles with virtual guidance, in *Proceedings of Chinese Control Conference*, 4283–4288, 2018.

[14] Y. Li, H. Guo, H. Gong, Y. Jiang, L. An, T. Ma, The improved adaptive hybrid fuzzy control of AUV horizontal motion, in *Proceedings of International Computer Conference on Wavelet Active Media Technology and Information Processing*, 408–414, 2016.

[15] Y. Hu, J. Yu, H. Yu, L. Zhao, C. Fu, Adaptive fuzzy command filtered control with error compensation mechanism for AUVs via backstepping, in *Proceedings of Chinese Control and Decision Conference*, 1226-1230, 2018.

[16] Y. Jiang, C. Guo, H. Yu, Adaptive trajectory tracking control for an underactuated AUV based on command filtered backstepping£¬in *Proceedings of Chinese Control Conference*, 1–5, 2018.

[17] J. Gonzalez, A. Benezra, S. Gomariz, Limitations of linear control for Cormoran-AUV, in *Proceedings of International Instrumentation and Measurement Technology Conference*, 1726–1729, 2012.

[18] Y. Jiang, C. Guo, H. Yu, Horizontal trajectory tracking control for an underactuated AUV adopted global integral sliding mode control£¬in *Proceedings of Chinese Control and Decision Conference*, 1–5, 2018.

[19] Y. Zhang, L. Gao, W. Liu, L. Li, Research on control method of AUV terminal sliding mode variable structure, in *Proceedings of International Conference on Robotics and Automation Sciences*, 88–93, 2017.

[20] S. Konar, M. Patil, V. Vyawahare, Design of a fractional order sliding mode controller for depth control of AUV, in *Proceedings of International Conference on Intelligent Computing and Control Systems*, 1342–1345, 2018.

[21] R. Cui, C. Yang, Y. Li, S. Sharma, Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning, *IEEE Transactions On Systems, Man, And Cybernetics: Systems*, 47: 1019–1029, 2017.

[22] B. Miao, T. Li, W. Luo, A DSC and MLP based robust adaptive NN tracking control for underwater vehicle, *Neurocomputing*, 184–189, 2013.

[23] H. Zhang, Y. Xu, J. Zhou, H. Fan, On the trajectory tracking problem of underactuated deep diving AUV, in *Proceedings of Chinese Control Conference*, 3812–3817, 2018.

[24] N. Gharesi, Z. Ebrahimi, A. Forouzandeh, M. Arefi, Extended state observer-based backstepping control for depth tracking of the underactuated AUV, in *Proceedings of International Conference on Control, Instrumentation, and Automation*, 354–358, 2017.

[25] J. Cervantes, W. Yu, S. Salazar, I. Chairez, Rogelio lozano output based backstepping control for trajectory tracking of an autonomous underwater vehicle, in *Proceedings of American Control Conference (ACC)*, 3512–3516, 2016.

[26] W. Xu, H. Li, J. Zhang, Y. Zhu, H. Zhang, Trajectory tracking for underwater rescue salvage based on backstepping control, in *Proceedings of Chinese Control Conference*, 2019.

[27] X. Peng, Research status and development of underwater robot, *Robot Technique and Application*, 15(4): 43–47, 2004.

[28] C. Petres, Y. Pailhas, P. Patron, Path planning for autonomous underwater vehicles, *IEEE Transactions on Robotics*, 23(2): 331–341, 2007.

Challenges of 3D NAND Flash Storage for Cyber-Physical Systems

Shuo-Han Chen, Academia Sinica

1 Introduction

NAND flash memory is widely regarded as a great storage medium for cyber-physical systems with its high read/write speed, small size, and shock resistance. In the past few years, 3D NAND flash technology has gathered the attention from both industry and academia as a high-density memory technology to increase storage density and cost-effectiveness of flash storage devices. The developing trend of 3D NAND flash well suit the high storage capacity and low-cost needs of today's cyber-physical systems. Comparing to the conventional planar (i.e., 2D) NAND flash, 3D NAND flash hugely improves the storage density of NAND flash devices by stacking memory cell vertically. The vertical structure allows 3D NAND flash to have high capacity without facing the downscaling issues, such as endurance, insufficient programming/erasing efficiency, and interference coupling issue, of 2D NAND flash. However, owing to the special vertical structure, 3D NAND flash reveals new challenges for the management flash devices. These challenges include (1) layer-to-layer process variation and (2) transient V_{th} shift phenomenon.

2 Structure of 3D NAND Flash

As stated in previous work [1], the minimal thickness of the tunnel oxide in 2D NAND flash should be at least 6 nm for preventing serious charge leakage and retaining enough data retention time. To overcome this issue, the charge-trap flash memories, such as SONOS and TANOS, has been proposed [2, 3, 4, 5] and regarded as the next-generation candidate for flash memory storage devices. Based on charge-trap material, building the charge trap flash memories in a 3D structure has also gathered increasing attention due to its high density and reasonable cost. Nevertheless, instead of vertically stacking planar flash memory cells, 3D charge trap flash memories are built with the "vertical channel." These vertical-channel 3D NAND flash memories include BiCS [6], TCAT [7], and SGVC [8]. The manufacturing process and the structure of 3D NAND flash can be summarized as Figure 1.

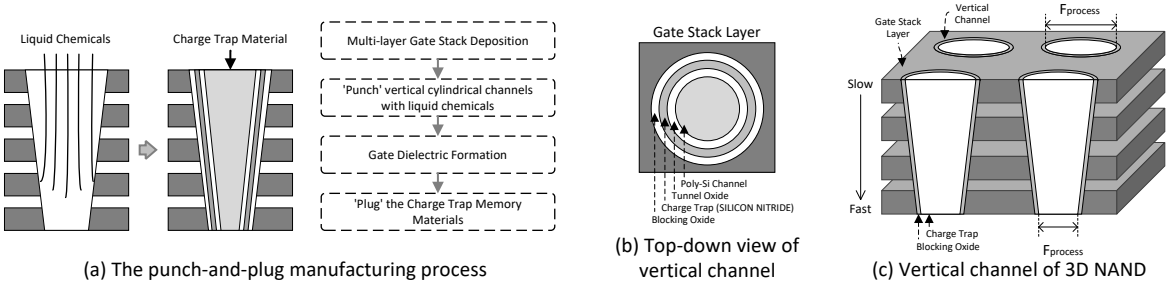

Figure 1: The manufacturing process and structure of 3D NAND flash [12].

As shown in Figure 2(a), the manufacturing process of vertical-channel 3D NAND flash is known as "punch-and-plug" method, in which liquid chemicals are utilized to erode through multiple gate stack layers. Owing to the physical characteristic of liquid chemicals, the created cylindrical channel will have a bigger opening at the top layer and a smaller opening at the bottom layer. In the end, this physical phenomenon results in asymmetric feature process size across the gate stack layers. Then, instead of filling floating gate materials, each cylindrical channel is filled with charge trap materials, as shown in Figure 2(b), to store bits at each gate stack layer. Finally, the structure of 3D NAND flash can be visualized as shown in Figure 2(c), the vertical channel 3D charge trap flash involves several vertical cylindrical channels and gate stack layers. Notably, the vertical channels of 3D NAND flash are regarded as blocks, while the channel sections located at each gate stack layer are mapped as pages [10, 11]. Meanwhile, even though both the planar NAND flash and 3D charge trap flash use Fowler-Nordheim Tunneling (FN) for conducting the program and erase operations, the voltage for programming or erasing 3D charge trap flash memory is lower than

that of planar NAND flash, thus resulting in less energy consumption and inducing less wear to the flash memory cells [14].

Notably, although 3D NAND flash memory has been developed, its distinct characteristics impose new challenges for the internal management, also known as flash translation layer (FTL) (e.g., FAST [9]), on flash devices. This is because flash memory has few inherent constraints, including the asymmetric access/erase units, erase-before-write property, and the limited number of program/erase (P/E) cycles. Notably, the block, which contains a fixed number of pages, is the basic unit of erase operations, and pages are the unit of access operations, including read and write operations. In addition, due to these inherent constraints, FTL needs to perform garbage collection and wear leveling mechanisms for reclaiming invalid pages and ensuring flash pages will not wear out prematurely. In the end, on top of these constraints, the management challenges of the next-generation 3D NAND flash have emerged with 3D NAND flash's distinct characteristics.

3 Challenges of 3D NAND Flash Storage

In this section, the challenges of the emerging 3D NAND Flash will be introduced in two folds, including (1) layer-to-layer process variation and (2) transient V_{th} shift phenomenon. First, due to the layer-to-layer process variation, the process size of each pages in a 3D NAND flash block are different. Then, the asymmetric feature process size at each gate stack layer results in the different strengths of the electric field along the vertical channels. As shown in Figure 3, the large the opening is, the smaller the electric field will be. In addition, prior studies [15, 16] also show that the threshold voltage shift speed will increase and the program latency will lengthen as the electric field increases. Therefore, accessing bits stored in layers with higher electric field strength will have shorter latency than accessing those bits stored at layers with smaller electric field strength. Furthermore, as the number of gate stacked layers become higher, the access speed difference of the top and bottom layers will be come larger. According to Figure 2, the electric field strength of 3D charge trap memory could go upto 2× to 5× difference with 0.25 to 0.75 nm difference in hole diameter. As the vertical channel sections located at each gate stack layer are mapped as pages and the vertical channels are managed as blocks, pages of the same block will have inconsistent access speed due to the unique cylindrical shape of vertical channels. In practice, the irregular page access speed feature of 3D charge-trap flash can be exploited to enhance the performance of flash based storage [12]. On the other hand, the process variation also affect the reliability of each pages; thus, pages of a single block also have different bit error rates. Therefore, the FTL management should be carefully considered to deal with the layer-to-layer process variation in terms of access performance and reliability.

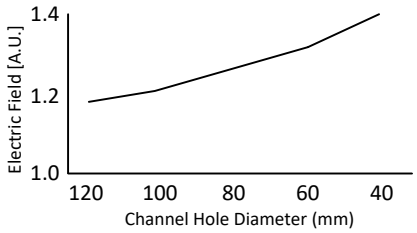

Figure 2: The electric field variation of 3D NAND flash.

Another distinct feature of 3D charge-trap NAND flash is the transient V_{th} shift phenomenon of performing erase operations on flash cells. As illustrated in Figure 3, the transient V_{th} shift phenomenon refers to the condition that the voltage of memory cells do not settle to a final value immediately after the flash cells is erased. Note that, in Figure 3, the gate voltage, V_{gate} , is initially connected to the ground before the erase voltage, V_{erase} , is applied to erase the cell content. Even though the actual causes of the transient V_{th} shift phenomenon is still under investigation by researchers, the negative impaces of this transient V_{th} shift phenomenon can already been observed during conducting the erase operation on 3D NAND flash devices. This is beasue the final voltage verification of an erase operation are delayed until the voltage of the erase flash cell settles to the final value after the transient V_{th} shift phenomenon. In addition, studies show the phenomenon of transient V_{th} shift actually become worse as the the number of P/E cycle

grows on flash devices [17]. Furthermore, according to previous investigation results, the latency of erasing flash cells on 3D charge trap NAND flash grows drastically as the P/E cycle grows larger than certain threshold (e.g., 10^4 [8]). In the end, the erase efficiency of 3D charge trap flash becomes worse and becomes a major concern in the garbage collection performance. This worsened erase efficiency also impose another management challenge for FTL management.

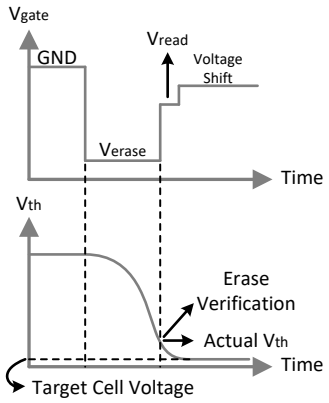

Figure 3: The transient V_{th} shift phenomenon of 3D NAND flash [13].

4 Conclusion

In this article, we summarize the structure and the management challenges of 3D NAND flash. Based on different characteristics of 3D NAND flash, various excellent studies have been proposed to either exploit the nice features or resolve the management difficulties of 3D NAND flash. Nevertheless, as the manufacturing process technology downscales, new challenges may surface and provide research opportunities for optimizing storage performance on cyber-physical systems.

References

[1] C. Zhao, C. Z. Zhao, S. Taylor, and P. R. Chalker. Review on nonvolatile memory with high-k dielectrics: Flash for generation beyond 32 nm. In Materials, 2014.

[2] J. Jang, H. S. Kim, W. Cho, H. Cho, J. Kim, S. I. Shim, Younggoan, J. H. Jeong, B. K. Son, D. W. Kim, Kihyun, J. J. Shim, J. S. Lim, K. H. Kim, S. Y. Yi, J. Y. Lim, D. Chung, H. C. Moon, S. Hwang, J. W. Lee, Y. H. Son, U. I. Chung, and W. S. Lee. Vertical cell array using tcat(terabit cell array transistor) technology for ultra high density nand flash memory. In 2009 Symposium on VLSI Technology, June 2009.

[3] F. R. Libsch and M. H. White. Charge transport and storage of low programming voltage sonos/monos memory devices. In Solid-State Electron, 1990.

[4] K. T. Park, J. m. Han, D. Kim, S. Nam, K. Choi, M. S. Kim, P. Kwak, D. Lee, Y. H. Choi, K. M. Kang, M. H. Choi, D. H. Kwak, H. w. Park, S. w. Shim, H. J. Yoon, D. Kim, S. w. Park, K. Lee, K. Ko, D. K. Shim, Y. L. Ahn, J. Park, J. Ryu, D. Kim, K. Yun, J. Kwon, S. Shin, D. Youn, W. T. Kim, T. Kim, S. J. Kim, S. Seo, H. G. Kim, D. S. Byeon, H. J. Yang, M. Kim, M. S. Kim, J. Yeon, J. Jang, H. S. Kim, W. Lee, D. Song, S. Lee, K. H. Kyung, and J. H. Choi. 19.5 three-dimensional 128gb mlc vertical nand flash-memory with 24-wl stacked layers and 50mb/s highspeed programming. In 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), Feb 2014.

[5] H. Tanaka, M. Kido, K. Yahashi, M. Oomura, R. Katsumata, M. Kito, Y. Fukuzumi, M. Sato, Y. Nagata, Y. Matsuoka, Y. Iwata, H. Aochi, and A. Nitayama. Bit cost scalable technology with punch and plug process for ultra high density flash memory. In 2007 IEEE Symposium on VLSI Technology, June 2007.

[6] H. Tanaka, M. Kido, K. Yahashi, M. Oomura, R. Katsumata, M. Kito, Y. Fukuzumi, M. Sato, Y. Nagata, Y. Matsuoka, Y. Iwata, H. Aochi, and A. Nitayama. Bit cost scalable technology with punch and plug process for ultra high density flash memory. In 2007 IEEE Symposium on VLSI Technology, June 2007.

[7] J. Jang, H. S. Kim, W. Cho, H. Cho, J. Kim, S. I. Shim, Younggoan, J. H. Jeong, B. K. Son, D.W. Kim, Kihyun, J. J. Shim, J. S. Lim, K. H. Kim, S. Y. Yi, J. Y. Lim, D. Chung, H. C. Moon, S. Hwang, J. W. Lee, Y. H. Son, U. I. Chung, and W. S. Lee. Vertical cell array using tcat(terabit cell array transistor) technology for ultra high density nand flash memory. In 2009 Symposium on VLSI Technology, June 2009.

[8] H. T. Lue, T. H. Hsu, C. J. Wu, W. C. Chen, T. H. Yeh, K. P. Chang, C. C. Hsieh, P. Y. Du, Y. H. Hsiao, Y. W. Jiang, G. R. Lee, R. Lo, Y. R. Su, C. Huang, S. C. Lai, L. Y. Liang, C. F. Chen, M. F. Hung, C. W. Hu, C. J. Chiu, and C. Y. Lu. A novel doubledensity, single-gate vertical channel (sgvc) 3d nand flash that is tolerant to deep vertical etching cd variation and possesses robust read-disturb immunity. In 2015 IEEE International Electron Devices Meeting (IEDM), pages 3.2.1–3.2.4, Dec 2015.

[9] S.-W. Lee, D.-J. Park, T.-S. Chung, D.-H. Lee, S. Park, and H.-J. Song. A log buffer-based flash translation layer using fully-associative sector translation. In ACM Transactions on Embedded Computing Systems (TECS), 2007.

[10] S.-H. Chen, Y.-T. Chen, H.-W. Wei, and W.-K. Shih. Boosting the performance of 3d charge trap nand flash with asymmetric feature process size characteristic. In Proceedings of the 54th Annual Design Automation Conference 2017, DAC '17, pages 83:1–83:6, 2017.

[11] R. Micheloni. 3D Flash Memories. Springer Nature, 2016.

[12] S. Chen, Y. Chen, Y. Chang, H. Wei and W. Shih, A Progressive Performance Boosting Strategy for 3-D Charge-Trap NAND Flash, in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2322-2334, Nov. 2018.

[13] S. Chen, Y. Chang, Y. Liang, H. Wei and W. Shih, An Erase Efficiency Boosting Strategy for 3D Charge Trap NAND Flash, in IEEE Transactions on Computers, vol. 67, no. 9, pp. 1246-1258, 1 Sept. 2018.

[14] A. Gupta, Y. Kim, and B. Urgaonkar. DFTL: a flash translation layer employing demand-based selective caching of page-level address mappings. In ASPLOS, 2009.

[15] J. H. Lee, G. S. Lee, S. Cho, J.-G. Yun, and B.-G. Park. Investigation of field concentration effects in arch gate silicon–oxide–nitride–oxide–silicon flash memory. In Japanese Journal of Applied Physics, 2010.

[16] S. D. Suk, K. H. Yeo, K. H. Cho, M. Li, Y. y. Yeoh, K. H. Hong, S. H. Kim, Y. H. Koh, S. Jung, W. Jang, D. W. Kim, D. Park, and B. I. Ryu. Gate-all-around twin silicon nanowire sonos memory. In 2007 IEEE Symposium on VLSI Technology, pages 142–143, June 2007.

[17] C.-P. Chen, H. T. Lue, C.-C. Hsieh, K.-P. Chang, K. Y. Hsieh, and C. Y. Lu. Study of fast initial charge loss and it's impact on the programmed states vt distribution of charge-trapping nand flash In 2010 International Electron Devices Meeting, pages 5.6.1–5.6.4, Dec 2010.

Use of Wearable Devices in Health Monitoring: A Review of Recent Studies

Ganapati Bhat, Ranadeep Deb, Umit Y. Ogras
School of ECEE, Arizona State University, Tempe, AZ
Email: {gmbhat, rdeb2, umit}@asu.edu

1 Introduction

About 15% of the world's population lives with a disability according to the annual world report on disability [45]. Moreover, 100 to 190 million individuals face significant difficulties in functioning. For instance, about 70 million people suffer from movement disorders, such as Parkinson's disease (PD), essential tremor (ET), epilepsy, and stroke [14]. State-of-the-art methodologies for diagnosis, treatment, and rehabilitation of this population rely on evaluations by medical professionals in a clinical environment [19]. However, as soon as patients leave the clinic it is not possible to monitor their symptoms due to lack of standard approaches [19]. Recent work suggests that *wearable internet-of-things (IoT) devices* that combine sensing, processing, and wireless communication can help in improving the quality of life of this population [38, 18, 22, 32].

Wearable sensors and mobile health applications are emerging as attractive solutions to augment clinical treatment and enable telepathic diagnostics [19, 14, 1]. Wearable devices have been recently used for monitoring of patients in a free-living home environment [28]. This capability allows doctors to understand the progression of symptoms over time [12]. Wearable devices have also shown promising results in the diagnosis and management of many movement disorders. For example, studies in [44, 36] use wearable sensors and machine learning algorithms to identify ET in patients. Similarly, Ryvlin et al. employ wearable devices to identify biomarkers that enable detection of generalized tonic-clonic seizures in patients with epilepsy [41]. Wearable technology has also been widely used in the diagnosis and treatment of PD patients [48, 11]. Despite these promising results, widespread adoption of wearable sensors and devices has been limited. Instead, they have been primarily used in research studies that occur in a more controlled environment.

This review summarizes how wearable devices are used in health monitoring. Specifically, it overviews the use of wearable devices in diagnosis, monitoring, and rehabilitation of movement disorders. Then, it discusses the major challenges and potential solutions to the adoption of wearable devices.

2 Wearable Devices in Health Monitoring

State-of-the-art methods for assessment and treatment of movement disorders are based on the tests performed in clinical examinations during which patients perform specific tasks. Diagnosis and evaluation of disease progression by visual inspection is sub-optimal as it can be affected by subjectivity of the clinician. Therefore, recent studies have explored the usage of wearable devices in the diagnosis and treatment of movement disorders [39, 37, 43]. To study the trends in the usage of wearable devices for movement disorders, we present a review of recent research in the following application areas: Diagnosis, prognosis/monitoring, predicting response to treatment, and therapy/rehabilitation.

Diagnosis/Early Diagnosis: Recent research on the diagnosis of movement disorders focuses mainly on assessing gait and tremor as these are some of the most commonly observed symptoms. The work in [47] employs an accelerometer to differentiate PD patients with gait disorder from healthy controls. Similarly, Zhang et al. [48] use data from wearable accelerometers and electromyography (EMG) sensors to develop a posture assessment system to differentiate between the tremor observed in ET and PD. Raethjen et al. study corticomuscular coherence of Parkinsonian tremor with the help of electroencephalogram (EEG) and EMG sensors [37]. Smartphones have also been used to aid in the diagnosis of ET and PD [44].

In addition to gait and tremor, many studies focus on diagnosis using non-motor symptoms, such as speech disorders and sleep disorders. For instance, Campos-Roca et al. use an acoustic data set from 40 healthy subjects and

40 PD patients to classify PD patients from the control subjects [10]. In summary, research on movement disorder diagnosis using wearable devices focuses on one of the following problems:

- Early diagnosis of PD patients
- Differentiate patients with PD from healthy controls or patients with a different neurological disorder
- Differentiating tremors caused by PD and ET

Prognosis/Monitoring the Severity of Symptoms: Objective measures are required for analysis of disease progression in patients since feedback from diaries and memory is subject to low compliance and recall bias [31]. Therefore, recent research has considered the following problems related to monitoring of patients:

- *Home-based or remote monitoring of patients:* Bächlin et al. developed a wearable system that uses accelerometers to detect freezing of gait events in PD patients [2]. Similarly, the study in [4] uses an accelerometer and a smartphone to analyze gait, dyskinesia, and motor states in real-time. Pulliam et al. [36] propose a system for continuous in-home monitoring of ET. A system for the detection of seizures in epilepsy patients is proposed in [5].
- *Evaluating the progression of movement disorders in patients:* Contreras et al. [12] propose a system that uses smartphones to analyze the severity of tremors in PD patients. The authors conclude that the proposed system can be used to evaluate the progression of PD in stages 3 and 4. Non-motor symptoms have also been used to monitor the progression of PD [29]. These studies use sensor data to analyze the emotional states of patients to better understand non-motor symptoms, such as anxiety and depression.
- *Evaluating the severity of symptoms in a patient:* Symptoms, such as tremor, can greatly affect the quality of life of patients. Therefore, recent studies have used wearable devices to evaluate the severity of tremor. Specifically, the works in [12, 36] use wearable sensors, such as gyroscope, to measure tremor amplitudes. For instance, Pulliam et al. use motion sensors to quantify the intensity of tremor in ET patients. Furthermore, the authors in [49] use four inertial sensors to assess bradykinesia and hypokinesia in PD patients.

Predicting Response to Treatment: Levodopa is one of the most commonly used medication to manage the symptoms of PD [40]. The dosage of Levodopa required varies as a function of the severity of symptoms. Currently, doctors measure the efficacy of treatment based on patients' diaries and observations. However, these inputs are highly subjective. Therefore, researchers employ wearable devices to analyze how patients are responding to treatment [34, 40]. Ruanola et al. [40] use recordings from a wireless EMG sensor mounted at the forearm to measure the effect of Levodopa in advanced PD patients. Wearable devices are also being used to understand how patients react to treatment in ET. Specifically, wearable technology can be used to monitor side-effects of ET treatment, such as heart problems [24].

Therapy and Rehabilitation: Wearable devices are also used for physiotherapy and other types of feedback in patients. For example, auditory cues and vibration-based devices have been used to help patients who experience freezing of gait [11, 43]. These methods are useful in alleviating symptoms like tremor and freezing of gait. For instance, Chomiak et al. [11] employ the sensors in an iPod touch to calculate the step height in walking. This data is then used to trigger auditory feedback to patients. Their results show that such a system is useful for stepping in place training. Similarly, Vidya et al. [43] use vibration motors on patients' wrists to reduce hand tremor. In summary, these studies show that wearable devices can be used effectively for therapy and rehabilitation in patients.

The study in [15] classifies 778 articles related to PD into one of the four application areas above. Figure 1 shows the composition of application areas for the 778 papers. We observe that the highest percentage (37%) of papers focus on diagnosis or assisting in the diagnosis of Parkinson's Disease. This is followed by the "Prognosis/Monitoring the Severity of Symptoms" category that has 36% of the papers. The other two application areas have a lower percentage of studies in the period 2008–2018. Specifically, 18% of the studies focus on therapy and rehabilitation of patients and 9% of the studies are classified in the category of predicting response to treatment. The lower number can be attributed to the fact that it is generally harder to predict the response to treatment as it requires monitoring as well.

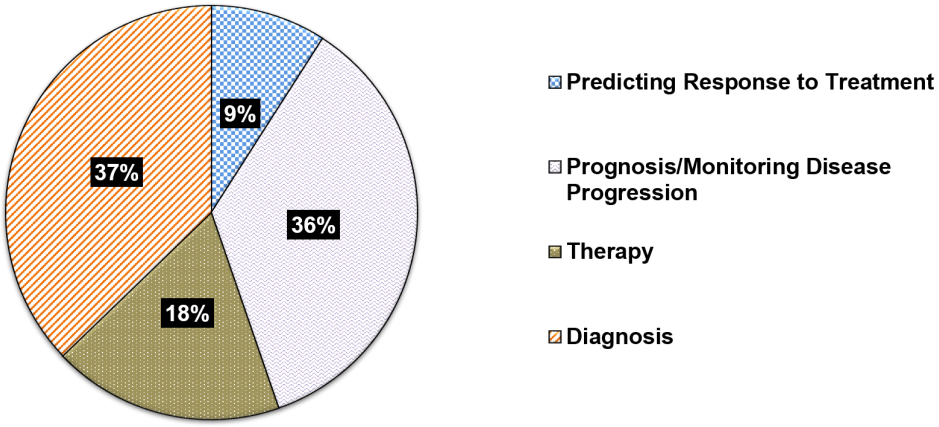

Figure 1: Percentage of publications on Parkinson's disease between 2008-2018 by application area

In summary, this review shows that wearable devices are making a high impact on movement disorders research by enabling patients and doctors to obtain more objective measures for the diagnosis and improving home monitoring of patients.

3 Wearable Challenges and Solutions Proposed in the Literature

The PD case study shows that wearable devices offer a great potential to improve the quality of life of patients. However, the current application of wearable devices is mainly limited to research studies. Recent research has focused on identifying the reasons that hinder the widespread adoption of wearable devices despite their potential in improving healthcare [21, 19, 23, 31]. The International Movement Disorders Task force states that non-compatible platforms, and limited applicability of "big data" acquired from the wearable devices as some of the reasons for lack of adoption [19]. Another study by Ozanne et al. [31] reports that many social and technical issues contribute to limited adoption by patients. According to surveys in [31], participants fear that bulky and rigid devices may lead to unwanted attention and a feeling of being watched. Instead, users prefer devices that are stretchable and flexible such that they can easily be worn under clothes. The surveys also highlight technical challenges, such as inconclusive recordings, privacy of data, and need for frequent recharging. Similarly, the review in [23] describes the needs of participants with various movement disorders. PD patients have typically expressed a need for wearable devices to assist in physiotherapy, while epilepsy patients want features that improve seizure management. The study in [27] states that about 32% of users stop using wearable devices after 6 months and addressing some of the issues above can enable a higher adoption rate for these devices. Therefore, a significant amount of research efforts focus on addressing one or more of these challenges [30, 1, 6]. In what follows, we overview recent research in the following areas:

1. Wearable IoT devices using Flexible Hybrid Electronics,
2. Energy-neutral operation through optimal energy harvesting and management,
3. New wearable applications that provide meaningful data to their users.

3.1 Wearable IoT Devices using Flexible Hybrid Electronics (FHE)

One of the major challenges faced by existing wearable devices is that they are typically rigid, which leads to patients stopping their use after a few weeks or months [31]. Flexible and stretchable electronics is emerging as an attractive technology to enable wearable devices. They can be used in applications such as electronics shirts and jackets [9].

However, the performance of pure flexible electronics is still much lower than that of conventional CMOS technologies. Flexible hybrid electronics has emerged as an attractive solution to bridge the gap between flexible electronics and CMOS technology [20, 26]. FHE technology has been used in developing health monitoring devices. For instance, Poliks et al. [35] use it to propose a wearable EEG monitor that can monitor a user, process the signal and transmit the data to a host. The work in [46] proposes a skin temperature monitor and an electrocardiogram (ECG) sensor using the FHE technology. We use FHE technology to propose an open-source wearable device that can help alleviate non-compatibility among wearable devices [6]. Our device integrates a TI-CC2650 microcontroller, inertial motion units, and multiple communication protocols to enable monitoring of movement disorders. We envision that the wearable prototype and extensions to it can be used to create an open-source hardware/software ecosystem for health monitoring by bringing health professionals and researchers together.

3.2 Energy-Neutral Operation

Energy limitation is one of the major challenges faced by wearable IoT devices. Large and inflexible batteries are not suitable for wearable use, whereas flexible printed batteries have limited capacities. Moreover, frequent recharging is cumbersome for patients suffering from functional disability [19]. Therefore, ensuring a long lifetime is one of the most critical requirements for the success of wearable devices. Dagdeviren et al. [13] propose a piezoelectric generator that is able to harvest energy from movements of the heart, lung, and diaphragm. This device can be easily integrated into a wearable health monitoring device to harvest energy from the human body. Similarly, solar energy harvesting for wearable devices has been studied in [33]. Ambient energy harvesting necessitates the development of algorithms to efficiently manage the harvested energy such that device lifetime can be maximized. To this end, Kansal et al. [25] propose the concept of energy-neutral operation where the energy by the device in any given period is equal to the harvested energy. Algorithms for energy-neutral operation are proposed in [25, 8]. These algorithms enable energy-neutral operation by maximizing the harvested energy and allocating it optimally.

3.3 Applications Areas for Wearable Devices

High impact applications using wearable devices are instrumental to the success of wearable devices [23]. Therefore, recent research has focused on developing meaningful applications using wearable devices. One of the most commonly implemented use cases is the monitoring of physical activity [17]. These devices help users in tracking their activity and in achieving fitness goals. Furthermore, human activity recognition has been a popular research area due to its applications in movement disorders. Human activity recognition using wearable devices has been proposed in [7, 3]. Another popular application is using wearable devices for vital sign monitoring, as surveyed in [16]. Sleep monitoring using wearable devices has also received attention recently due to its potential benefits in improving user wellness [42]. In conclusion, these application areas along with FHE technology and energy-neutral operation have the potential to significantly improve the adoption rates of wearable devices.

4 Conclusion

Wearable devices offer great potential to improve the quality of life for patients and the general population. This article presented a review of how wearable technology is being used in the diagnosis, monitoring, rehabilitation of movement disorder patients. Then, it discussed the major challenges that hinder the widespread adoption of wearable devices. Finally, it presented new proposals that aim to improve the adoption of wearable devices. Specifically, it focused on flexible hybrid electronics, energy-neutral operation, and health applications. We envision that these solutions will lead to wider adoption of wearable devices.

References

[1] I. Azimi, A. Anzanpour, A. M. Rahmani, T. Pahikkala, M. Levorato, P. Liljeberg, and N. Dutt, "Hich: Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT," *ACM Trans. on Embedd. Comput. Syst.*, vol. 16, no. 5s, p. 174, 2017.

[2] M. Bächlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Tröster, "Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom," *IEEE Trans. Information Technology in Biomedicine*, vol. 14, no. 2, pp. 436–446, 2010.

[3] L. Bao and S. S. Intille, "Activity Recognition From User-Annotated Acceleration Data," in *Int. Conf. on Pervasive Comput.*, 2004, pp. 1–17.

[4] À. Bayés *et al.*, "A "HOLTER" for Parkinson's disease: Validation of the ability to detect on-off states using the REMPARK system," *Gait & Posture*, vol. 59, pp. 1–6, 2018.

[5] S. Beniczky, T. Polster, T. W. Kjaer, and H. Hjalgrim, "Detection of Generalized Tonic–Clonic Seizures by a Wireless Wrist Accelerometer: A Prospective, Multicenter Study," *Epilepsia*, vol. 54, no. 4, pp. e58–e61, 2013.

[6] G. Bhat, R. Deb, and U. Y. Ogras, "OpenHealth: Open Source Platform for Wearable Health Monitoring," *IEEE Design Test*, pp. 1–1, 2019.

[7] G. Bhat *et al.*, "Online Human Activity Recognition using Low-Power Wearable Devices," in *Proc. ICCAD*, 2018.

[8] G. Bhat, J. Park, and U. Y. Ogras, "Near-Optimal Energy Allocation for Self-Powered Wearable Systems," in *Proc. ICCAD*, 2017, pp. 368–375.

[9] Q. Brogan, T. O'Connor, and D. S. Ha, "Solar and Thermal Energy Harvesting With a Wearable Jacket," in *Proc. IEEE Int. Symp. Circuits and Syst.*, 2014.

[10] Y. Camnos-Roca, F. Calle-Alonso, C. J. Perez, and L. Naranjo, "Computational Diagnosis of Parkinson's Disease from Speech Based on Regularization Methods," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1127–1131.

[11] T. Chomiak, A. Watts, N. Meyer, F. V. Pereira, and B. Hu, "A Training Approach to Improve Stepping Automaticity While Dual-Tasking in Parkinson's Disease: A Prospective Pilot Study," *Medicine*, vol. 96, no. 5, 2017.

[12] R. Contreras *et al.*, "Tremors Quantification in Parkinson Patients Using Smartwatches," in *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*. IEEE, 2016, pp. 1–6.

[13] C. Dagdeviren *et al.*, "Conformal Piezoelectric Energy Harvesting and Storage From Motions of the Heart, Lung, and Diaphragm," *Proceedings of the National Academy of Sciences*, vol. 111, no. 5, pp. 1927–1932, 2014.

[14] J.-F. Daneault, "Could Wearable and Mobile Technology Improve the Management of Essential Tremor?" *Frontiers in Neurology*, vol. 9, pp. 257:1–257:8, 2018.

[15] R. Deb, "How Does Technology Development Influence the Assessment of Parkinson's Disease? A Systematic Review," Master's thesis, Arizona State University, 2019.

[16] D. Dias and J. Paulo Silva Cunha, "Wearable Health Devices–Vital Sign Monitoring, Systems and Technologies," *Sensors*, vol. 18, no. 8, p. 2414, 2018.

[17] K. M. Diaz *et al.*, "Fitbit®: An Accurate and Reliable Device for Wireless Physical Activity Tracking," *Int. J. Cardiology*, vol. 185, pp. 138–140, 2015.

[18] D. V. Dimitrov, "Medical Internet of Things and Big Data in Healthcare," *Healthcare Informatics Research*, vol. 22, no. 3, pp. 156–163, 2016.

[19] A. J. Espay *et al.*, "Technology in Parkinson's Disease: Challenges and Opportunities," *Movement Disorders*, vol. 31, no. 9, pp. 1272–1282, 2016.

[20] U. Gupta, J. Park, H. Joshi, and U. Y. Ogras, "Flexibility-Aware System-on-Polymer (SoP): Concept to Prototype," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 3, no. 1, pp. 36–49, 2017.

[21] M. Hartmann, U. S. Hashmi, and A. Imran, "Edge Computing in Smart Health Care Systems: Review, Challenges, and Research Directions," *Trans. on Emerging Telecommunications Technologies*, p. e3710, 2019.

[22] S. Hiremath, G. Yang, and K. Mankodiya, "Wearable Internet of Things: Concept, Architectural Components and Promises for Person-Centered Healthcare," in *Proc. MOBIHEALTH*, 2014, pp. 304–307.

[23] D. Johansson, K. Malmgren, and M. A. Murphy, "Wearable Sensors for Clinical Applications in Epilepsy, Parkinson's Disease, and Stroke: A Mixed-Methods Systematic Review," *Journal of Neurology*, pp. 1–13, 2018.

[24] P. Kakria, N. Tripathi, and P. Kitipawang, "A Real-Time Health Monitoring System for Remote Cardiac Patients Using Smartphone and Wearable Sensors," *International Journal of Telemedicine and Applications*, vol. 2015, p. 8, 2015.

[25] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embedd. Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.

[26] Y. Khan *et al.*, "Flexible Hybrid Electronics: Direct Interfacing of Soft and Hard Electronics for Wearable Health Monitoring," *Advanced Functional Materials*, vol. 26, no. 47, pp. 8764–8775, 2016.

[27] D. Ledger, "Inside Wearables-Part 2," *Endeavour Partners*, 2014.

[28] S. I. Lee *et al.*, "Activity Detection in Uncontrolled Free-Living Conditions Using a Single Accelerometer," in *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2015, pp. 1–6.

[29] M. Memedi *et al.*, "Automatic Spiral Analysis for Objective Assessment of Motor Symptoms in Parkinson's Disease," *Sensors*, vol. 15, no. 9, pp. 23 727–23 744, 2015.

[30] A. A. Mutlag, M. K. A. Ghani, N. a. Arunkumar, M. A. Mohamed, and O. Mohd, "Enabling Technologies for Fog Computing in Healthcare IoT Systems," *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.

[31] A. Ozanne *et al.*, "Wearables in Epilepsy and Parkinson's Disease—-A Focus Group Study," *Acta Neurologica Scandinavica*, vol. 137, no. 2, pp. 188–194, 2018.

[32] A. Pantelopoulos and N. G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 40, no. 1, pp. 1–12, 2010.

[33] J. Park *et al.*, "Flexible PV-cell Modeling for Energy Harvesting in Wearable IoT Applications," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 5s, p. 156, 2017.

[34] P. H. Pelicioni *et al.*, "Head and Trunk Stability During Gait Before and After Levodopa Intake in Parkinson's Disease Subtypes," *Experimental Gerontology*, vol. 111, pp. 78–85, 2018.

[35] M. Poliks *et al.*, "A Wearable Flexible Hybrid Electronics ECG Monitor," in *Proc. Electron. Components and Tech. Conf.*, 2016, pp. 1623–1631.

[36] C. Pulliam *et al.*, "Continuous In-Home Monitoring of Essential Tremor," *Parkinsonism & Related Disorders*, vol. 20, no. 1, pp. 37–40, 2014.

[37] J. Raethjen *et al.*, "Cortical Correlates of the Basic and First Harmonic Frequency of Parkinsonian Tremor," *Clinical Neurophysiology*, vol. 120, no. 10, pp. 1866–1872, 2009.

[38] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-Physical Systems: The Next Computing Revolution," in *Design Automation Conference*. IEEE, 2010, pp. 731–736.

[39] E. Rovini, C. Maremmani, and F. Cavallo, "How Wearable Sensors Can Support Parkinson's Disease Diagnosis and Treatment: A Systematic Review," *Frontiers in Neuroscience*, vol. 11, p. 555, 2017.

[40] V. Ruonala *et al.*, "Levodopa-Induced Changes in Electromyographic Patterns in Patients With Advanced Parkinson's Disease," *Frontiers in Neurology*, vol. 9, p. 35, 2018.

[41] P. Ryvlin, C. Ciumas, I. Wisniewski, and S. Beniczky, "Wearable Devices for Sudden Unexpected Death in Epilepsy Prevention," *Epilepsia*, vol. 59, pp. 61–66, 2018.

[42] X. Sun, L. Qiu, Y. Wu, Y. Tang, and G. Cao, "Sleepmonitor: Monitoring Respiratory Rate and Body Position During Sleep Using Smartwatch," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, p. 104, 2017.

[43] V. Vidya, P. Poornachandran, V. Sujadevi, and M. M. Dharmana, "Suppressing Parkinson's Diseases Induced Involuntary Movements Using Wearables," in *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*. IEEE, 2017, pp. 1–4.

[44] A. M. Woods, M. Nowostawski, E. A. Franz, and M. Purvis, "Parkinson's Disease and Essential Tremor Classification on Mobile Device," *Pervasive and Mobile Computing*, vol. 13, pp. 1–12, 2014.

[45] World Health Organization, "World Report on Disability," [Online] http://www.who.int/disabilities/world_report/2011/report/en/.

[46] Y. Yamamoto *et al.*, "Efficient Skin Temperature Sensor and Stable Gel-Less Sticky ECG Sensor for a Wearable Flexible Healthcare Patch," *Advanced Healthcare Materials*, vol. 6, no. 17, p. 1700495, 2017.

[47] M. Yoneyama, Y. Kurihara, K. Watanabe, and H. Mitoma, "Accelerometry-Based Gait Analysis and Its Application to Parkinson's Disease assessment–Part 2: A New Measure for Quantifying Walking Behavior," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 6, pp. 999–1005, 2013.

[48] B. Zhang, F. Huang, J. Liu, and D. Zhang, "A Novel Posture for Better Differentiation Between Parkinson's Tremor and Essential Tremor," *Frontiers in Neuroscience*, vol. 12, 2018.

[49] D. G. Zwartjes, T. Heida, J. P. Van Vugt, J. A. Geelen, and P. H. Veltink, "Ambulatory Monitoring of Activities and Motor Symptoms in Parkinson's Disease," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 11, pp. 2778–2786, 2010.

A Taxonomy for Convolutional Neural Network Inference Acceleration

Qianru Zhang, Meng Zhang, Guoqing Li, Guodong Tong
Department of Electronic Science & Engineering, Southeast University, Nanjing, China

1 Introduction

Convolutional neural network (CNN) architectures have been around for over two decades. Compared with other neural network models such as multiple layer perceptron (MLP), CNN is designed to take multiple arrays as input and then process the input using convolution operator within a local field by mimicking eyes perceiving images. Therefore, it shows excellent performance in solving computer vision problems such as image classification, recognition, object detection and understanding [1, 2, 3, 4]. It is also effective for a wide range of fields such as autonomous driving [5], speech recognition that requires correlated speech spectral representations [6], VLSI physical design [7], multi-media compression [8] comparing with the traditional DCT transformation and compressive sensing methods [9, 10], and cancer detection from a series of condition changing images [11].

However, in order to receive good performance of prediction and accomplish more difficult goals, CNN architecture becomes deeper and more complicated. At the same time, more pixels are packed into one image thanks to high resolution acquisition devices. As a result, CNN training and prediction are very computationally expensive and become limited for implementation due to its slow speed. In many situations, CNN training is not as often as inference. Once the network is well-trained, inference is made every time the new input is given in a feed forward flow. Therefore, inference speed is critical when implementing the well-trained network. Although acceleration for CNN inference has been explored since it was brought up, recently this seems to be keener as it has such good industrial impact.

In this article, we review many recent works and summarize inference acceleration methods in software level and hardware level. Many methods improve the training efficiency, which results in the acceleration on the inference. That will also be covered in this work. This article is organized as following. In Section 2, an overview of modern CNN structure is given with the description of different typical layers. In Section 3 we present our taxonomy for recent CNN inference acceleration methods followed by the detailed inference acceleration methods in two levels summarized in Section 4 and in Section 5 respectively. Finally, Section 6 concludes this article with some future challenges.

2 Convolutional Neural Network

The modern convolutional neural networks proposed by LeCun [12] is a 7-layer (excluding the input layer) LeNet-5 structure. It has the following structure C1, S2, C3, S4, C5, F6, OUTPUT as shows in Fig. 1, where C indicates convolutional layer, S indicates subsampling layer, and F indicates fully-connected layer. There are many modifications regarding the structure of CNNs in order to handle more complicated datasets and problems, such as AlexNet (8 layers) [13], GoogLeNet (22 layers) [14], VGG-16 (16 layers) [15], and ResNet (152 layers) [16]. Table 1 summarizes the state-of-the-art CNNs. As we can see from the table, the number of parameters in modern CNNs is large, which usually takes a long time for training and for inference. Plus, higher dimensional input, large number of parameters, and complex CNN configuration challenge hardware in terms of processing element efficiency, memory bandwidth, off-chip memory, communication and so on [17].

Among these different structures, they share four key features including weight sharing, local connection, pooling, and the use of many layers [20]. Also, there are some commonly used layers such as convolutional layers, subsampling layers (pooling layers), and fully-connected layers. Usually, there is a convolutional layer after the input. The convolutional layer is often followed by a subsampling layer. This combination repeats several times to increase the depth of CNN. The fully-connected layers are designed as the last few layers in order to map from extracted features to labels. These four layers are introduced as follows.

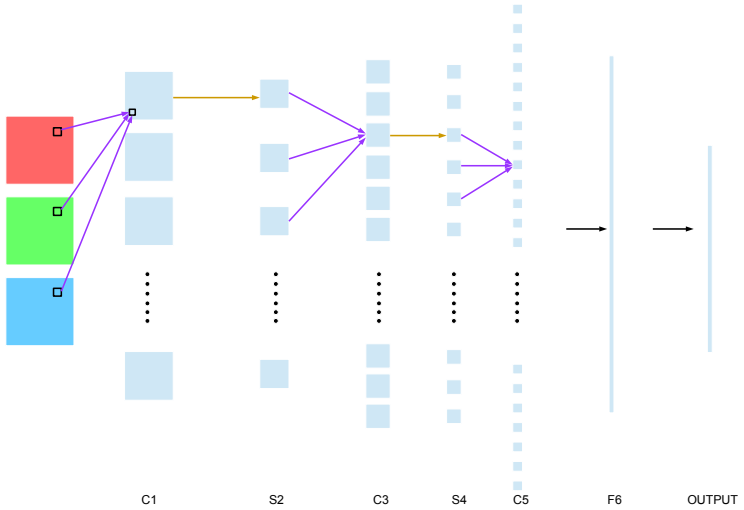

Figure 1: Illustration of LeNet-5.

Model	Layer Size	Configuration	Feature	Parameter Size	Application
LeNet [12]	7 layers	3C-2S-1F-RBF output layer		60,000	Document recognition
AlexNet [13]	8 layers	5C-3S-3F	Local response normalization	60,000,000	Image classification
NIN [18]	-	3mlpconv-global average pooling (S can be added in between the mlpconv)	mlpconv layer: 1C-3MLP; global average pooling	-	Image classification
VGG [15]	11-19 layers	VGG-16: 13C-5S-3F	Increased depth with stacked 3×3 kernels	133,000,000 to 144,000,000	Image classification and localization
ResNet [16]	Can be very deep (152 layers)	ResNet-152: 151C-2S-1F	Residual module	ResNet-20: 270,000; ResNet-1202: 19,400,000	Image classification, object detection
GoogLeNet [14]	22 layers	3C-9Inception-5S-1F	Inception module	6,797,700	Image classification, object detection
Xception [19]	37 layers	36C-5S-1F	Depth-wise separable convolutions	22,855,952	Image classification

Table 1: CNN model summary.
C: convolutional layer, S: subsampling layer, F: fully-connected layer

a) Input Layer: In CNNs, input layers usually take multiple arrays and are often size-fixed. Comparing to ordinary fully-connected neural networks, the CNN input do not need size-normalization and centralization. Because CNN enjoys the characteristic of translation invariance [21].

b) Convolutional Layer: As a key feature layer that makes CNNs different from other ordinary neural networks, neuron units of convolutional layers are first computed by convolution operation over small local patches of input, and then followed by activation functions (tanh, sigmoid, ReLU, etc.), and form a 2D feature map (3D feature map channel).

c) Subsampling Layer (pooling layer): Convolutional layers are usually followed by subsampling layers to reduce the feature map resolution. The amount of parameters and computation are also reduced accordingly.

d) Fully-connected Layer: After several layers, high-level features are extracted and require mapping to labels. In fully-connected layer, neuron units are transformed from 2D into 1D. Each unit in the current layer is connected to all the units in the previous layer such like regular neural networks. It not only extracts features in a more complex way in order to dig deep for more information, but patterns in different locations are connected as well.

e) Output Layer: As a feed-forward neural network, the output layer neuron units are fixed. They are usually linked with previous neurons in a fully-connected way and they are the final threshold for predicting.

In general, CNNs have gained a lot of interest in researching the meaning behind the combination of those different layers. The advantages brought by the structure of CNNs include reduced number of parameters and translation invariance.

3 Acceleration Method Taxonomy

Our taxonomy shows in Fig. 2. For the CNN structure, there is redundancy in both weights and the number of bits for representation. For the redundancy in weights, layer decomposition, network pruning, block-circulant projection and knowledge distillation methods can be applied. For the redundancy in representation, using fixed-point representation is the mainstream. Considering that convolutional layers are computationally intensive, we are also interested in the convolution operation complexity. Therefore, we also summarize some efficient convolution methods that are adopted in the CNN. As for the hardware level, the mainstream GPU, FPGA, ASIC are discussed. Recently, people see a promising future for fast implementation of CNN as neuromorphic engineering develops. Some new devices are also presented in this article. The acceleration approaches of each level is orthogonal and can be combined with those in other levels. By researching such a wide range of methods, we expect to provide a general idea on CNN acceleration.

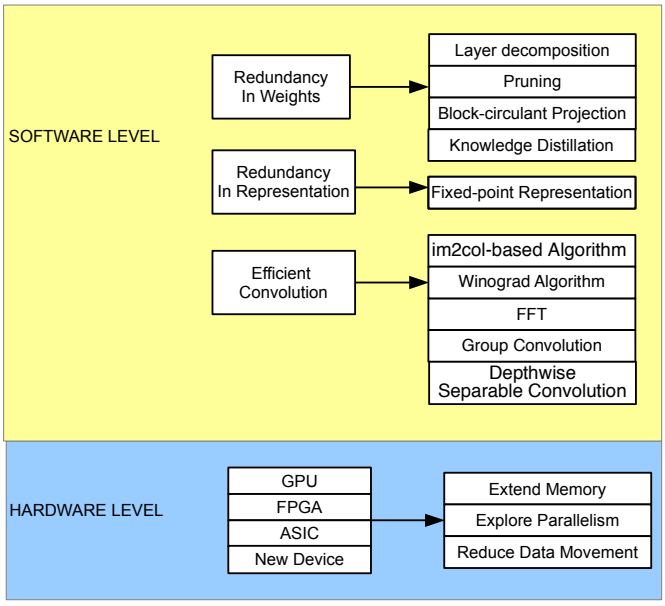

Figure 2: Taxonomy of CNN inference acceleration methods.

4 Software Level

The training and inference process can be accelerated by reducing redundancy in network structures. There is redundancy both in weights and in the way how weights are represented. Two perspectives of acceleration methods will be summarized as follows in terms of redundancy in weights and redundancy in representations. Also, by changing how convolution performs, such as using im2col-based Algorithm, Winograd Algorithm, Group Convolution, Depthwise Separable Convolution, or transforming into frequency domain, it can improve convolution speed and reduce memory consumption.

4.1 Redundancy In Weights

There is significant redundancy in the parameterization of some neural networks. As Denil *et al*. and Sainath *et al*. observe that some weights learned in networks are correlated with each other, they demonstrate that some of the weights can either be predicted or be unnecessary to learn [22, 23].

4.1.1 Layer Decomposition

Low-rank approximation can be adopted to reduce redundancy in weights [24, 25]. An efficient low-rank approximation of kernels can be applied in first few convolutional layers of CNN to exploit the linear structure of the over-parameterization within a filter. For example, Denton *et al*. reduce the computation work for redundancy within kernels. It achieves $2 \sim 2.5\times$ speedup with less than 1% drop in classification performance for a single convolutional layer. It uses singular value decomposition method to exploit the approximation of kernels with assumptions that the singular values of the kernels decay rapidly so that the size of the kernels can be reduced significantly [26].

Instead of treating kernel filters as different matrices, kernels in one layer can be treated as a 3D tensor with two spatial dimensions and the third dimension representing channels. Lebedev *et al*. use CP-decomposition for convolutional layers, which achieves $8.5\times$ CPU speedup at the cost of 1% error increase [27]. Tai *et al*. utilize tensor decomposition to remove the redundancy in the convolution kernels, which achieves twice more efficiency of inference for VGG-16 [28]. Wang *et al*. propose to use group sparse tensor decomposition for each convolutional layer, which achieves $6.6\times$ speed-up on PC and $5.91\times$ speed-up on mobile device with less than 1% error rate increase [29]. Tucker decomposition is also used recently to decompose pre-trained weights with fine-tuning afterwards [30, 31].

Weight matrix decomposition method can not only be applied to convolutional layers, but also fully-connected layers. Applying the low-rank approximation to the fully-connected layer weight can achieve a $30 \sim 50\%$ reduction of number of parameters with little loss in accuracy, which is roughly an equivalent reduction in training time [23].

The decomposition technique is layer oriented and can be interleaved with other modules such as ReLU modules in CNN. It can also be applied to the structure of neural networks. Rigamonti *et al*. apply this technique to the general frameworks and reduce the computational complexity by using linear combinations of fewer separable filters [32]. This method can be extended for multiple layers (e.g. > 10) by utilizing low-rank approximation for both weights and input [33]. It can achieve $4\times$ speedup with 0.3% error increase for deep network models VGG-16 by focusing on reducing accumulated error across layers using generalized singular value decomposition.

The methods above can be generalized as layer decomposition for filter weight matrix dimension reduction, while pruning is another method for dimension reduction.

4.1.2 Network Pruning

Network pruning originates as a method to reduce the size and over-fitting of a neural network. As neural network implementation on hardware becomes more popular, it is necessary to reduce the limitation such as its intensive computation and large memory bandwidth requirement. Nowadays, pruning is usually adopted as a method to reduce the network size and to increase the network inference speed so that it can be applied in specific hardware such as embedded systems.

There are many pruning methods in terms of weights, connections, filters, channels, feature maps, and so on. Unlike layer decomposition in which computational complexity is reduced through reducing the total size of layers, selected neurons are removed in pruning. For pruning weights, the unimportant connections of weights with magnitudes smaller than a given threshold are dropped. Experiments are taken on NVIDIA TitanX and GTX980 GPUs, which achieves 9× and 13× parameter reduction for AlexNet and VGG-16 models respectively with no loss of accuracy [34]. Zhou *et al*. incorporate sparse constraints to decimate the number of neurons during training, which reduces the 70% number of neurons without accuracy sacrifice [35]. Channel pruning method is to eliminate lowly active channels, which means filters are applied in fewer number of channels in each layer. Polyak *et al*. propose a channel-pruning based method Inbound Prune to compress a redundant network. Their experiment is taken on the platform of Samsung Galaxy S6 and it achieves 1.59× speedup [36]. Recently, pruning is combined with other acceleration techniques to achieve speedup. For example, Han *et al*. combine pruning with trained quantization and Huffman coding to deep compress the neural networks in three steps. It achieves 3× layer-wise speedup on fully-connected layer over benchmark on CPU [34].

Some of these pruning methods result in structured sparsity, while others cause unstructured sparsity such as weight-based pruning. Many techniques are proposed to deal with problems of unstructured sparsity being unfriendly to hardware. Wen *et al*. propose a method called Structured Sparsity Learning (SSL) for regularizing compressed structures of deep CNNs and speeding up convolutional computation by group Lasso regularization and locality optimization respectively. It improves convolutional layer computation speed by 5.1x and 3.1x over CPU and GPU [37]. He *el al*. propose a channel pruning method by iteratively reducing redundant channels through solving LASSO and reconstructing the outputs with linear least squares. It achieves 5x speed increase in VGG-16 and 2x speedup in ResNet/ Xception [38].

4.1.3 Block-circulant Projection

A square matrix could be represented by a one-block-circulant matrix, while a non-squared matrix could be represented by block-circulant matrix. Block-circulant based CNN has been explored nowadays as it has small storage requirements.

Cheng *et al*. apply the circulant matrix in the fully connected layer and achieve significant gain in efficiency with little decrease in accuracy [39]. Yang *et al*. focus on reducing the computational time spent in fully-connected layer by imposing the circulant structure on the weight matrix for dimension reduction with little loss in performance [40]. Ding *et al*. propose to use block-circulant structure in both fully-connected layers and convolutional layers in non-square-matrix situations to further reduce the storage waste. They also mathematically prove that fewer weights in circulant form do not harm the ability of a deep CNN without weight redundancy reduction [41].

4.1.4 Knowledge Distillation

Knowledge distillation is a concept that information obtained from a large complex ensemble neural networks can be utilized to form a compact neural network [42]. The way that knowledge is transferred can be depicted in the following Fig. 3. Information flow from one complex network to a simpler one by training the latter one with data labeled by the former network. By using synthetic data generated from a complex network to train a compact model, it is less likely to cause overfitting and can approximate the functions very well. More importantly, it provides a new perspective for model compression and complicated neural network acceleration.

Bucilu *et al*. lay a foundation for mimicking a large machine learning model by experimenting three ways to generate pseudo data, which are random, naive bayes estimation, and MUNGE respectively [43]. Some researches propose teacher-student format, which also adopts knowledge distillation concepts with different methods for synthesizing data. For example, Hinton *et al*. compress a deep teacher network into a student network using data combined from teacher network outcome and the true labeled data [44].

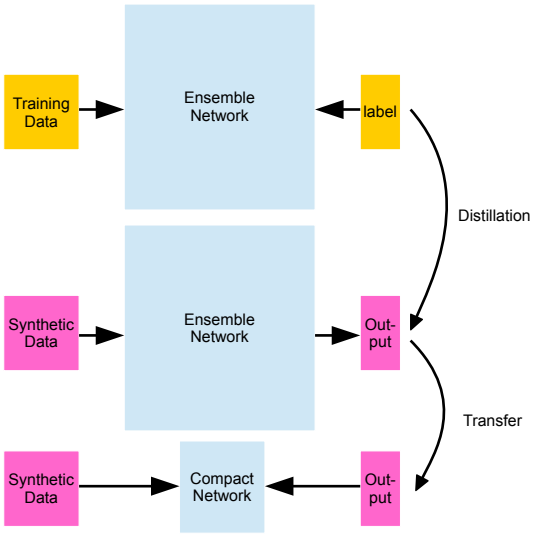

Figure 3: Illustration of knowledge distillation

4.2 Redundancy in Representations

Many weights in neural networks have very small values. Most arithmetic operations in neural networks use 32-floating point representation in order to achieve a good accuracy. As a trade-off, that increases the computation workload and memory size for the neural networks. However, arithmetic operations in fixed-point instead of floating-point can achieve enough good performance for neural networks [45]. A 16-bit fixed-point representation method is proposed by using stochastic rounding for training CIFAR-10 dataset [46]. A further compression of 10-bit dynamic fixed-point is also explored [47]. Han *et al*. quantize pruned CNNs to 8-bit and achieve further storage reduction with no loss of accuracy [34]. In terms of binarization, there are many works focusing on binary input, binary weights of the network, and binary operations [48, 49, 50, 51, 52, 53]. Ternary CNNs are proposed recently as a more expressive method comparing to binary CNNs, which seeks to achieve a balance between binary networks and full precision networks in terms of compression rate and accuracy [54, 55, 56].

Stochastic computing (SC) is a type of technique that simplifies numerical computations into bit-wise operations by representing continuous values with random bit streams. It provides many benefits for neural networks such as low computation footprint, error tolerance, simple implementation in circuits and better trade-off between time and accuracy [57]. Many works contribute to exploring potential space in optimization and in deep belief networks [58, 59, 60]. Recently it starts to gain attentions in CNN field and regarded as a promising technique for deep CNN implementation [61, 62].

Although errors may accumulate due to representation approximation, its hardware implementation can achieve a much faster speed and lead to less energy consumption.

4.3 Efficient Convolution

4.3.1 im2col-based Algorithm

For the direct convolution in the CNN, convolution kernels slide over the two dimensions of the input and the output is obtained by dot product between the kernels and the input. While for the im2col-based algorithms, the input matrix is linearized into multiple lowered vectors, which can be later efficiently computed [63, 64, 65]. Cho *et al*. further reduce the linearization memory-overhead and improve the computational efficiency by modifying both the lowered vectors and the vectorized kernels [66].

4.3.2 Winograd Algorithm

Winograd based methods are to incorporate Winograd's minimal filtering algorithms to compute minimal convolution over small filters. Coppersmith Winograd algorithm is known as a fast matrix multiplication algorithm. Winograd based convolution reduces the multiplications by increasing the number of additions and it reduces the memory consumption on GPU [67, 68]. Winograd's minimal filtering algorithms can help reduce convolution computation at the expense of memory bandwidth. Xiao *et al* utilize Winograd's minimal filtering theory combined with heterogeneous algorithms for a fusion architecture to mitigate memory bandwidth problem [69].

4.3.3 FFT

Based on the experiment that FFT can be applied in MLP to accelerate the first layer inference [70], Mathieu *et al*. first apply FFT on weights of CNN and achieve good performance and speedup for large perceptive areas [71].

The concept of implementing CNN in frequency is to replace convolution operation in time domain with multiplication in frequency domain. It takes time to transform back and forth. As a result, it performs well on large feature maps. Development is made to suit for small feature maps such as training network directly in frequency domain. Compared with other algorithms, FFT method requires additional memory for padding the filters to the same size of the input and storing frequency domain intermediate results. This leads to a trade-off for hardware implementation. On one hand, it can take use of power in GPU parallelism to speedup convolution computation dramatically. On the other hand, more delicate GPU memory allocation is required due to limit memory.

4.3.4 Group Convolution

Group convolution is first applied to AlexNet for incorporating two GPUs working together [13]. The input feature maps are partitioned into groups and within each group, regular convolution is implemented. Recently, some works focus on CNN acceleration using group convolution [72, 73, 74, 75], which not only reduces the convolution computation but also can improve the performance.

4.3.5 Depthwise Separable Convolution

A standard convolution can be decomposed into a depthwise convolution and a pointwise convolution. For the depthwise convolution, it can extract the spatial information of one feature map. On the other hand, the pointwise convolution fuses the information across different channels. By manipulating the separable convolution, new CNN structures can be created, which can achieve high accuracy [76] or reduce many parameters [77].

5 Hardware Level

Neural networks regain their vigor due to high performance hardware recently. CPU used to be the main stream for implementing machine learning algorithms about twenty years ago, because matrix multiplication and factorization techniques were not popular back then. Nowadays, GPU, FPGA, and ASIC are utilized for accelerating training and predicting process. Besides, much new device technology is proposed to meet requirement for very large models and large training datasets. In the following, hardware based accelerators are summarized in terms of GPU, FPGA, ASIC and frontier new device that is promising for accelerating deep convolutional neural networks.

5.1 GPU

In terms of GPU, clusters of GPUs can accelerate very large neural networks with over one billion parameters in a parallel way. The mainstream of GPU cluster neural networks usually work with distributed SGD algorithms as illustrated in Fig. 4. Many researches further exploit the parallelism and make efforts on communication among different clusters. For example, Baidu Heterogeneous Computing Group uses two types of parallelism called model-data parallelism and data parallelism to extend CNN architectures to 36 servers, each with 4 NVIDIA Tesla K40m

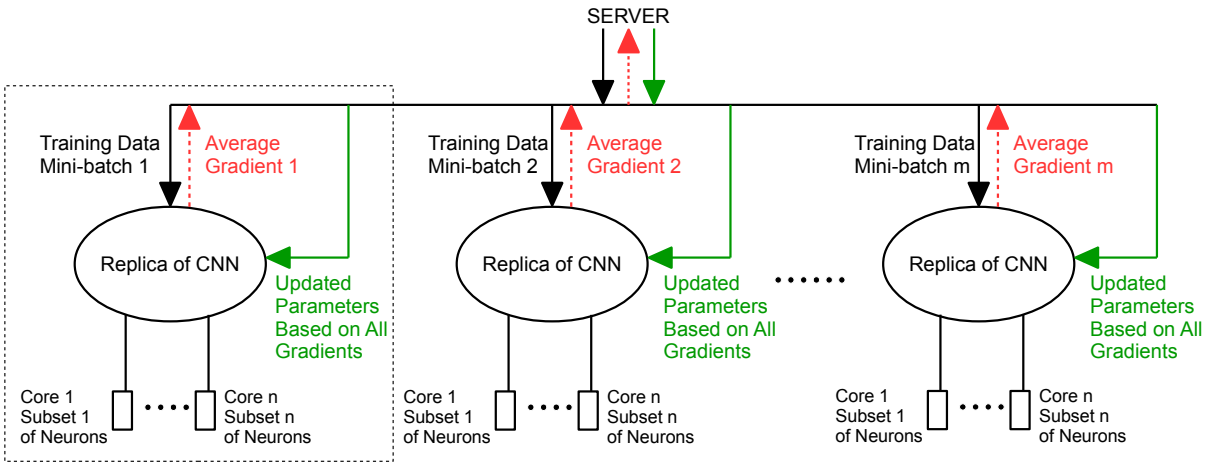

Figure 4: Illustration of CNN distributed system.

GPUs and 12GB memory. The strategies include butterfly synchronization and lazy update, which makes good use of overlapping in computation and communication [78]. Coates *et al*. propose a clustering of GPU servers using Commodity Off-The-Shelf High Performance Computing (COTS HPC) technology and high-speed communication infrastructure for parallelism in distributed gradient decent algorithm, which reduces 98% number of machines used for training [79]. In terms of non-distributed SGD algorithms, Imani *et al*. propose a nearest content addressable memory block called NNCAM, which stores highly frequent patterns for reusing. It accelerates CNNs over general purpose GPU with 40% speedup [80].

GPU supports several teraFLOPS throughput and large memory access, but consumes a lot of energy. In terms of economy, GPU costs to set up for large deep convolutional neural networks.

5.2 ASIC

For ASIC design, despite of using methods in software level such as block-circulant projection in Section 4.1.3 and SC in Section 4.2, memory can be expanded and locality can be increased to reduce data transporting within systems for deep neural network accelerating. Tensor Processing Unit (TPU) is designed for low precision computation with high efficiency. It uses a large on-chip memory of 28MiB to execute the neural network applications, which can achieve at most 30× faster speed than an Nvidia K80 GPU [81]. TETRIS is an architecture using 3D memory proposed by Gao *et al*. It saves more area for processing elements and leaves more space for accelerator design [82].

Luo *et al*. create an architecture of 64-chip system that minimizes data moving between synapses and neurons by storing them closely. It reduces the burden on external memory bandwidth and achieves a speedup of 450× over a GPU with 150× energy reduction [83]. Wang *et al*. propose to group adjacent process engines (PEs) into dual-channel PEs called Chain-NN to mitigate huge amount of data movements. They simulate it under TSMC 28nm process and achieve a peak throughput of 806.4 GOPS in AlexNet [84]. Single instruction multiple data (SIMD) processors are used on a 32-bit CPU to design a system targeted for ASIC synthesis to perform real-time detection, recognition and segmentation of mega-pixel images. They optimize the operation in CNN with available parallelism in hardware. The ASIC implementations outperform the CPU conventional methods in terms of frames/s [85].

Recently, some ASIC designs target for sparse networks with irregularity. For example, Zhang *et al*. propose an accelerator called Cambricon-X. It consists an Indexing Module, which can efficiently schedule processing elements that store irregular and compressed synapses. The accelerator can reach 544 GOP/s in 6.38mm^2 [86]. Kwon *et al*. design a reconfigurable accelerator called MAERI to adapt various layer dataflow patterns. They can efficiently utilize compute resources and provides 6.9× speedup at 50% sparsity [87]. Network pruning could induce sparsity and irregularity as discussed in Section 4.1.2. With such designs, better performance is expected to achieve when combined.

Comparing to GPU, ASIC is specialized hardware and can be delicately designed to maximize its benefits such

as power-efficiency and large throughput in CNN implementation. However, once CNN algorithms are implemented on ASIC, it is difficult to change the hardware design. On the other hand FPGA is easy to be programmed and reconfigured. It is more convenient for prototyping.

5.3 FPGA

There are many parallelism levels in hardware acceleration, such as coarse-grain, medium-grain, fine-grain, and massive [88]. FPGA outperforms in terms of its fine grain and coarse grain reconfiguration ability and its hierarchical storage structure and scheduling mechanism can be optimized flexibly. Flexible hierarchical memory systems can support complex data access mode of CNN. It is often used to improve the efficiency of on-chip memory and to reduce the energy consumption in deep neural network implementation [89].

Peemen *et al*. experiment on Virtex 6 FPGA board and show that the accelerator design can achieve 11× speedup with very complicated address mapping of data access [90]. Zhang *et al*. take data reuse, parallel processing, and off-chip memory bandwidth into consideration in FPGA accelerator. The accelerator achieves 17.42× faster speed than CPU in AlexNet CNN architecture [91]. Martinez *et al*. take advantage of the FPGA reconfiguration characteristics by unfolding the loop execution on different cascading stages. As the number of multipliers for convolution increases, the proposed method can achieve 12 GOPS at most [92]. A hardware acceleration method for CNN is proposed by combining fine grain in operator level parallelism and coarse grain parallelism. Compared with 4xIntel Xeon 2.3 GHz, 1.35 GHz C870, and a 200 MHz FPGA, the proposed design achieves a 4× to 8× speed boost [93]. Wang *et al*. propose an on-chip memory design called Memsqueezer that can be implemented on FPGA. They shrink the memory size by compressing data, weights, and intermediate data from the perspectives of hardware, which achieves 80% energy reduction compared with conventional buffer designs [94]. Zhang *et al*. design an FPGA accelerator engine called Caffeine that decreases underutilized memory bandwidth. It reorganizes the memory access according to their proposed matrix-multiplication representation applied to both convolutional layers and fully-connected layers. Caffeine's implementation on Xilinx KU060 and Virtex 7690t FPGA achieves very high peak performance of 365 GOPS and 636 GOPS respectively [95]. Rahman *et al*. present a 3D array architecture, which can benefit all layers in CNNs. With optimization of on-chip buffer sizes for FPGAs, it can outperform the state-of-the-art solutions by 22% in terms of MAC [96]. Alwani *et al*. explore the design space of dataflow across multiple convolutional layers, where a fused layer accelerator is designed that reduces feature map data transfer from and to off-chip memory [97].

Compared with GPU, FPGA throughput is tens of gigaFLOPS and it has limited memory access. Plus, it does not support floating-point natively. But it is more power-efficient. Due to its limited memory access, many proposed methods are focused on accelerating inference time of neural network since inference process requires less memory access comparing to training process. Others are emphasized on external memory optimization for large neural network acceleration. Different models need different hardware optimization and even for the same model, different designs result in quite various acceleration performance [34]. In terms of economy, FPGA is reconfigurable and is easier to evolve hardware, frameworks and software. Especially for various models of neural networks, its flexibility shortens design cycle and costs less.

5.4 New Devices

As new device technology and circuits arise, deep convolutional neural networks can be potentially accelerated by orders of magnitude. In terms of new device, very large scale integration systems are explored to mimic complex biological neuron architectures.

Some of them are in their theoretical demonstration state for training deep neural networks. For example, Gokmen and Vlasov from IBM research center propose a resistive processing unit (RPU) device, which can both store and compute parameters in this unit. It has extremely high processing speed with 30000× higher than state-of-art microprocessors (84000 GigaOps/s/W) [98]. As neuromorphic engineering develops, more new device emerges to handle high frequency and high volume information transformation through synapses. Some are in theoretical state that have not been implemented on neural networks for classification and recognition, such as nano-scale phase change device [99] and ferroelectric memristors [100].

Resistive memories are treated as one of the promising solutions for deep neural network accelerations due to its nonvolatility, high storage density, and low power consumption [101]. Its architecture mimics neural networks, where weight storage and computation can be done simultaneously [102, 103]. As CMOS memories become larger, its scale becomes limited. Therefore, besides the main stream CMOS based memory, nonvolatile memory becomes more popular in storing weights, such as resistive random access memory (RRAM) [104, 105, 106, 107] and spin-transfer torque random access memory (STT-RAM) [108].

Memristor crossbar array structure can deal with computational expensive matrix multiplication and have been explored in CNN hardware implementations. For example, Hu *et al*. develop a Dot-Product Engine (DPE) utilizing memristor crossbar, which achieves 1000× to 10,000× speed-efficiency product compared with a digital ASIC [109]. Xia *et al*. address energy consumption problem between crossbars and ADC/DAC and can save more than 95% energy with similar accuracy of CNN [110]. Ankit *et al*. propose a hierarchical reconfigurable architecture with memristive crossbar arrays called RESPARC. It is 15× more energy efficient and has 60× more throughput for deep CNNs [111].

In general, for any CNN hardware implementation, there are a lot of potential solutions to be explored in design space. It is not trivial to design a general hardware architecture that can be applied to every CNN, especially when limitations on computation resource and memory bandwidth are considered.

6 Conclusion

In this article, we summarize the recent advances in CNN Inference acceleration methods in terms of software level and hardware level. In software level, CNN is compressed without losing significant accuracy since there is redundancy in most of the CNN architectures. Convolution calculation is also an important factor for CNN. FFT method introduces a frequency perspective for training neural networks. In hardware level, characteristics for different hardware such as FPGA and GPU are explored combined with CNN features. CNN performs better in computer vision field as its structure goes deeper and the amount of data becomes larger, which makes it time consuming and computationally expensive. It is imperative and necessary to accelerate CNN for its further implementation in life. For now, there is no generalized evaluation system to test the acceleration performance for comparison among different methods in different levels. Researches use case by case dataset benchmark and different criterion in each level. Therefore, it is challenging in acceleration performance evaluation as well.

References

[1] S. Yu, S. Jia, C. Xu, Convolutional neural networks for hyperspectral image classification, Neurocomputing 219 (2017) 88–98.

[2] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8) (2013) 1915–1929.

[3] X. Xu, X. Zhang, B. Yu, X. S. Hu, C. Rowen, J. Hu, Y. Shi, Dac-sdc low power object detection challenge for uav applications, arXiv preprint arXiv:1809.00110.

[4] K. Wang, C. Gou, N. Zheng, J. M. Rehg, F.-Y. Wang, Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives, Artificial Intelligence Review 48 (3) (2017) 299–329.

[5] H. Zhou, W. Li, Y. Zhu, Y. Zhang, B. Yu, L. Zhang, C. Liu, Deepbillboard: Systematic physical-world testing of autonomous driving systems, arXiv preprint arXiv:1812.10812.

[6] P. Qin, W. Xu, J. Guo, An empirical convolutional neural network approach for semantic relation classification, Neurocomputing 190 (2016) 1–9.

[7] B. Yu, D. Z. Pan, T. Matsunawa, X. Zeng, Machine learning and pattern matching in physical design, in: 20th Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2015, pp. 286–293.

[8] L. Theis, W. Shi, A. Cunningham, F. Huszár, Lossy image compression with compressive autoencoders, in: International Conference on Learning Representations, 2017.

[9] M. Zhang, T. Chen, X. Shi, P. Cao, Image arbitrary-ratio down-and up-sampling scheme exploiting dct low frequency components and sparsity in high frequency components, IEICE Transactions on Information and Systems 99 (2) (2016) 475–487.

[10] T. Chen, M. Zhang, J. Wu, C. Yuen, Y. Tong, Image encryption and compression based on kronecker compressed sensing and elementary cellular automata scrambling, Optics & Laser Technology 84 (2016) 118–133.

[11] J. A. A. Jothi, V. M. A. Rajam, A survey on automated cancer diagnosis from histopathology images, Artificial Intelligence Review 48 (1) (2017) 31–81.

[12] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.

[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[17] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, B. Yu, Recent advances in convolutional neural network acceleration, Neurocomputing 323 (2019) 37–51.

[18] M. Lin, Q. Chen, S. Yan, Network In Network, ArXiv e-printsarXiv:1312.4400.

[19] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[20] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[21] C. M. Bishop, Pattern recognition and machine learning, Springer, New York, 2006.

[22] M. Denil, B. Shakibi, L. Dinh, N. de Freitas, et al., Predicting parameters in deep learning, in: Advances in Neural Information Processing Systems, 2013, pp. 2148–2156.

[23] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, B. Ramabhadran, Low-rank matrix factorization for deep neural network training with high-dimensional output targets, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6655–6659.

[24] M. Jaderberg, A. Vedaldi, A. Zisserman, Speeding up convolutional neural networks with low rank expansions, in: Proceedings of the British Machine Vision Conference. BMVA Press, 2014.

[25] Y. Ma, R. Chen, W. Li, F. Shang, W. Yu, M. Cho, B. Yu, A Unified Approximation Framework for Compressing and Accelerating Deep Neural Networks, arXiv e-prints (2018) arXiv:1807.10119arXiv:1807.10119.

[26] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, in: Advances in Neural Information Processing Systems, 2014, pp. 1269–1277.

[27] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, V. Lempitsky, Speeding-up convolutional neural networks using fine-tuned cp-decomposition, in: International Conference on Learning Representations, 2014.

[28] C. Tai, T. Xiao, Y. Zhang, X. Wang, W. E, Convolutional neural networks with low-rank regularization, 2016.

[29] P. Wang, J. Cheng, Accelerating convolutional neural networks for mobile applications, in: Proceedings of the ACM on Multimedia Conference, ACM, 2016, pp. 541–545.

[30] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin, Compression of deep convolutional neural networks for fast and low power mobile applications, arXiv preprint arXiv:1511.06530.

[31] H. Ding, K. Chen, Y. Yuan, M. Cai, L. Sun, S. Liang, Q. Huo, A compact cnn-dblstm based character model for offline handwriting recognition with tucker decomposition, in: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Vol. 1, IEEE, 2017, pp. 507–512.

[32] R. Rigamonti, A. Sironi, V. Lepetit, P. Fua, Learning separable filters, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[33] X. Zhang, J. Zou, K. He, J. Sun, Accelerating very deep convolutional networks for classification and detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (10) (2016) 1943–1955.

[34] S. Han, H. Mao, W. J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, in: International Conference on Learning Representations, 2016.

[35] H. Zhou, J. M. Alvarez, F. Porikli, Less is more: Towards compact cnns, in: European Conference on Computer Vision, Springer, 2016, pp. 662–677.

[36] A. Polyak, L. Wolf, Channel-level acceleration of deep face representations, IEEE Access 3 (2015) 2163–2175.

[37] W. Wen, C. Wu, Y. Wang, Y. Chen, H. Li, Learning structured sparsity in deep neural networks, in: Advances in Neural Information Processing Systems, 2016, pp. 2074–2082.

[38] Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks, in: International Conference on Computer Vision (ICCV), Vol. 2, 2017, p. 6.

[39] Y. Cheng, F. X. Yu, R. S. Feris, S. Kumar, A. Choudhary, S. F. Chang, An exploration of parameter redundancy in deep networks with circulant projections, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2857–2865.

[40] Z. Yang, M. Moczulski, M. Denil, N. de Freitas, A. Smola, L. Song, Z. Wang, Deep fried ConvNets, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1476–1483.

[41] C. Ding, S. Liao, Y. Wang, Z. Li, N. Liu, Y. Zhuo, C. Wang, X. Qian, Y. Bai, G. Yuan, et al., C ir cnn: accelerating and compressing deep neural networks using block-circulant weight matrices, in: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, ACM, 2017, pp. 395–408.

[42] R. Caruana, A. Niculescu Mizil, G. Crew, A. Ksikes, Ensemble selection from libraries of models, in: Proceedings of the 21st International Conference on Machine Learning, ACM, 2004, p. 18.

[43] C. Buciluǎ, R. Caruana, A. Niculescu Mizil, Model compression, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 535–541.

[44] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531.

[45] D. Hammerstrom, A vlsi architecture for high-performance, low-cost, on-chip learning, in: IEEE International Joint Conference on Neural Networks, IEEE, 1990, pp. 537–544.

[46] S. Gupta, A. Agrawal, K. Gopalakrishnan, P. Narayanan, Deep learning with limited numerical precision., in: Proceedings of The 32nd International Conference on Machine Learning, 2015, pp. 1737–1746.

[47] M. Courbariaux, Y. Bengio, J. P. David, Training deep neural networks with low precision multiplications, arXiv preprint arXiv:1412.7024.

[48] M. Courbariaux, Y. Bengio, J. P. David, Binaryconnect: Training deep neural networks with binary weights during propagations, in: Advances in Neural Information Processing Systems, 2015, pp. 3123–3131.

[49] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, Xnor-net: Imagenet classification using binary convolutional neural networks, in: European Conference on Computer Vision, Springer, 2016, pp. 525–542.

[50] M. Kim, P. Smaragdis, Bitwise neural networks, in: Proceedings of The 33rd International Conference on Machine Learning, 2016.

[51] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, Y. Zou, Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients, arXiv preprint arXiv:1606.06160.

[52] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Quantized neural networks: Training neural networks with low precision weights and activations, arXiv preprint arXiv:1609.07061.

[53] H. Kim, J. Sim, Y. Choi, L.-S. Kim, A kernel decomposition architecture for binary-weight convolutional neural networks, in: Proceedings of the 54th Annual Design Automation Conference, ACM, 2017, p. 60.

[54] F. Li, B. Zhang, B. Liu, Ternary weight networks, arXiv preprint arXiv:1605.04711.

[55] Z. Lin, M. Courbariaux, R. Memisevic, Y. Bengio, Neural networks with few multiplications, 2016.

[56] H. Alemdar, V. Leroy, A. Prost-Boucle, F. Pétrot, Ternary neural networks for resource-efficient ai applications, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2547–2554.

[57] B. D. Brown, H. C. Card, Stochastic neural computation. i. computational elements, IEEE Transactions on computers 50 (9) (2001) 891–905.

[58] Z. Li, A. Ren, J. Li, Q. Qiu, B. Yuan, J. Draper, Y. Wang, Structural design optimization for deep convolutional neural networks using stochastic computing, in: Proceedings of the Conference on Design, Automation & Test in Europe, European Design and Automation Association, 2017, pp. 250–253.

[59] K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, K. Choi, Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks, in: Proceedings of the 53rd Annual Design Automation Conference, ACM, 2016, p. 124.

[60] Y. Ji, F. Ran, C. Ma, D. J. Lilja, A hardware implementation of a radial basis function neural network using stochastic logic, in: Proceedings of the Design, Automation & Test in Europe Conference & Exhibition, EDA Consortium, 2015, pp. 880–883.

[61] A. Ren, Z. Li, C. Ding, Q. Qiu, Y. Wang, J. Li, X. Qian, B. Yuan, Sc-dcnn: highly-scalable deep convolutional neural network using stochastic computing, in: Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, ACM, 2017, pp. 405–418.

[62] J. Li, Z. Yuan, Z. Li, A. Ren, C. Ding, J. Draper, S. Nazarian, Q. Qiu, B. Yuan, Y. Wang, Normalization and dropout for stochastic computing-based deep convolutional neural networks, Integration, the VLSI Journal.

[63] cuBLAS, http://docs.nvidia.com/cuda/cublas.

[64] MLK, https://software.intel.com/en-us/intel-mkl.

[65] OpenBLAS, http://www.openblas.net.

[66] M. Cho, D. Brand, Mec: Memory-efficient convolution for deep neural network, in: International Conference on Machine Learning, 2017, pp. 815–824.

[67] A. Lavin, S. Gray, Fast algorithms for convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4013–4021.

[68] H. Park, D. Kim, J. Ahn, S. Yoo, Zero and data reuse-aware fast convolution for deep neural networks on gpu, in: International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS), IEEE, 2016, pp. 1–10.

[69] Q. Xiao, Y. Liang, L. Lu, S. Yan, Y.-W. Tai, Exploring heterogeneous algorithms for accelerating deep convolutional neural networks on fpgas, in: 54th ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2017, pp. 1–6.

[70] S. Ben Yacoub, B. Fasel, J. Luettin, Fast face detection using mlp and fft, in: Proceedings of the 2nd International Conference on Audio and Video-based Biometric Person Authentication, 1999, pp. 31–36.

[71] M. Mathieu, M. Henaff, Y. LeCun, Fast training of convolutional networks through ffts, arXiv preprint arXiv:1312.5851.

[72] S. Xie, R. Girshick, P. Dollar, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[73] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[74] T. Zhang, G.-J. Qi, B. Xiao, J. Wang, Interleaved group convolutions, in: The IEEE International Conference on Computer Vision (ICCV), 2017.

[75] G. Li, M. Zhang, b. Duan, Q. Zhang, G. Tong, Kernel sharing in the channel dimension to improve parameters efficiency, in: Proceedings of the 2019 International Conference on Computing, Electronics & Communications Engineering, 2019.

[76] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[77] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861.

[78] R. Wu, S. Yan, Y. Shan, Q. Dang, G. Sun, Deep image: Scaling up image recognition, arXiv preprint arXiv:1501.02876 7 (8).

[79] A. Coates, B. Huval, T. Wang, D. Wu, B. Catanzaro, N. Andrew, Deep learning with cots hpc systems, in: Proceedings of The 30th International Conference on Machine Learning, 2013, pp. 1337–1345.

[80] M. Imani, D. Peroni, Y. Kim, A. Rahimi, T. Rosing, Efficient neural network acceleration on gpgpu using content addressable memory, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2017, pp. 1026–1031.

[81] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, et al., In-datacenter performance analysis of a tensor processing unit, arXiv preprint arXiv:1704.04760.

[82] M. Gao, J. Pu, X. Yang, M. Horowitz, C. Kozyrakis, Tetris: Scalable and efficient neural network acceleration with 3d memory, in: Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems, ACM, 2017, pp. 751–764.

[83] T. Luo, S. Liu, L. Li, Y. Wang, S. Zhang, T. Chen, Z. Xu, O. Temam, Y. Chen, Dadiannao: A neural network supercomputer, IEEE Transactions on Computers 66 (1) (2017) 73–88.

[84] S. Wang, D. Zhou, X. Han, T. Yoshimura, Chain-nn: An energy-efficient 1d chain architecture for accelerating deep convolutional neural networks, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2017, pp. 1032–1037.

[85] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, E. Culurciello, Hardware accelerated convolutional neural networks for synthetic vision systems, in: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, IEEE, 2010, pp. 257–260.

[86] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, Y. Chen, Cambricon-x: An accelerator for sparse neural networks, in: 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), IEEE, 2016, pp. 1–12.

[87] H. Kwon, A. Samajdar, T. Krishna, Maeri: Enabling flexible dataflow mapping over dnn accelerators via reconfigurable interconnects, in: Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems, ACM, 2018, pp. 461–475.

[88] N. Izeboudjen, C. Larbes, A. Farah, A new classification approach for neural networks hardware: from standards chips to embedded systems on chip, Artificial Intelligence Review 41 (4) (2014) 491–534.

[89] Q. Sun, T. Chen, J. Miao, B. Yu, Power-driven dnn dataflow optimization on fpga, in: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2019.

[90] M. Peemen, A. A. Setio, B. Mesman, H. Corporaal, Memory-centric accelerator design for convolutional neural networks, in: IEEE 31st International Conference on Computer Design, IEEE, 2013, pp. 13–19.

[91] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, J. Cong, Optimizing FPGA-based accelerator design for deep convolutional neural networks, in: Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ACM, 2015, pp. 161–170.

[92] J. J. Martínez, J. Garrigós, J. Toledo, J. M. Ferrández, An efficient and expandable hardware implementation of multilayer cellular neural networks, Neurocomputing 114 (2013) 54–62.

[93] S. Chakradhar, M. Sankaradas, V. Jakkula, S. Cadambi, A dynamically configurable coprocessor for convolutional neural networks, in: Proceedings of IEEE International Symposium on Circuits and Systems, ACM, 2010.

[94] Y. Wang, H. Li, X. Li, Re-architecting the on-chip memory sub-system of machine-learning accelerator for embedded devices, in: Proceedings of the 35th International Conference on Computer-Aided Design, ACM, 2016, p. 13.

[95] C. Zhang, Z. Fang, P. Zhou, P. Pan, J. Cong, Caffeine: Towards uniformed representation and acceleration for deep convolutional neural networks, in: IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2016, pp. 1–8.

[96] A. Rahman, J. Lee, K. Choi, Efficient fpga acceleration of convolutional neural networks using logical-3d compute array, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2016, pp. 1393–1398.

[97] M. Alwani, H. Chen, M. Ferdman, P. Milder, Fused-layer cnn accelerators, in: 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), IEEE, 2016, pp. 1–12.

[98] T. Gokmen, Y. Vlasov, Acceleration of deep neural network training with resistive cross-point devices: Design considerations, Frontiers in Neuroscience 10.

[99] B. L. Jackson, B. Rajendran, G. S. Corrado, M. Breitwisch, G. W. Burr, R. Cheek, K. Gopalakrishnan, S. Raoux, C. T. Rettner, A. Padilla, et al., Nanoscale electronic synapses using phase change devices, ACM Journal on Emerging Technologies in Computing Systems (JETC) 9 (2) (2013) 12.

[100] S. Saïghi, C. G. Mayr, T. Serrano Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier, S. Boyn, A. F. Vincent, D. Querlioz, et al., Plasticity in memristive devices for spiking neural networks, Frontiers in Neuroscience 9 (2015) 51.

[101] J. Seo, B. Lin, M. Kim, P. Y. Chen, D. Kadetotad, Z. Xu, A. Mohanty, S. Vrudhula, S. Yu, J. Ye, et al., On-chip sparse learning acceleration with CMOS and resistive synaptic devices, IEEE Transactions on Nanotechnology 14 (6) (2015) 969–979.

[102] X. Zeng, S. Wen, Z. Zeng, T. Huang, Design of memristor-based image convolution calculation in convolutional neural network, Neural Computing and Applications (2016) 1–6.

[103] Y. Shim, A. Sengupta, K. Roy, Low-power approximate convolution computing unit with domain-wall motion based "spin-memristor" for image processing applications, in: 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2016, pp. 1–6.

[104] L. Ni, H. Huang, H. Yu, On-line machine learning accelerator on digital rram-crossbar, in: IEEE International Symposium on Circuits and Systems, IEEE, 2016, pp. 113–116.

[105] Z. Xu, A. Mohanty, P. Y. Chen, D. Kadetotad, B. Lin, J. Ye, S. Vrudhula, S. Yu, J. Seo, Y. Cao, Parallel programming of resistive cross-point array for synaptic plasticity, Procedia Computer Science 41 (2014) 126–133.

[106] M. Prezioso, F. Merrikh Bayat, B. Hoskins, G. Adam, K. K. Likharev, D. B. Strukov, Training and operation of an integrated neuromorphic network based on metal-oxide memristors, Nature 521 (7550) (2015) 61–64.

[107] M. Cheng, L. Xia, Z. Zhu, Y. Cai, Y. Xie, Y. Wang, H. Yang, Time: A training-in-memory architecture for memristor-based deep neural networks, in: 54th ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2017, pp. 1–6.

[108] L. Song, Y. Wang, Y. Han, H. Li, Y. Cheng, X. Li, Stt-ram buffer design for precision-tunable general-purpose neural network accelerator, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25 (4) (2017) 1285–1296.

[109] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, R. S. Williams, Dot-product engine for neuromorphic computing: programming 1t1m crossbar to accelerate matrix-vector multiplication, in: Proceedings of the 53rd annual design automation conference, ACM, 2016, p. 19.

[110] L. Xia, T. Tang, W. Huangfu, M. Cheng, X. Yin, B. Li, Y. Wang, H. Yang, Switched by input: Power efficient structure for rram-based convolutional neural network, in: 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2016, pp. 1–6.

[111] A. Ankit, A. Sengupta, P. Panda, K. Roy, Resparc: A reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks, in: Proceedings of the 54th Annual Design Automation Conference, ACM, 2017, p. 27.

1 Conferences and Workshops

- IEEE International Conference on Cyber Physical and Social Computing (CPSCom 2019)
- DAC-2019 Workshop on Design Automation for Cyber-Physical Systems (DACPS-2019)
- IEEE International Conference on Embedded Software and Systems (ICESS)
- IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)

2 Special Issues in Academic Journals

- IEEE/CAA Journal of Automatica Sinica special issue on Resilient Control in Large-Scale Networked Cyber-Physical Systems

Call for Contributions

Newsletter of Technical Committee on Cyber-Physical Systems (IEEE Systems Council)

The newsletter of Technical Committee on Cyber-Physical Systems (TC-CPS) aims to provide timely updates on technologies, educations and opportunities in the field of cyber-physical systems (CPS). The letter will be published twice a year: one issue in February and the other issue in October. We are soliciting contributions to the newsletter. Topics of interest include (but are not limited to):

- Embedded system design for CPS
- Real-time system design and scheduling for CPS
- Distributed computing and control for CPS
- Resilient and robust system design for CPS
- Security issues for CPS
- Formal methods for modeling and verification of CPS
- Emerging applications such as automotive system, smart energy system, internet of things, biomedical device, etc.

Please directly contact the editors and/or associate editors by email to submit your contributions.

Submission Deadline:

All contributions must be submitted by **Jan. 15, 2020** in order to be included in the February issue of the newsletter.

Editors:

- Bei Yu, Chinese University of Hong Kong, Hong Kong, byu@cse.cuhk.edu.hk

Associate Editors:

- Xianghui Cao, Southeast University, China, xhcao@seu.edu.cn
- Long Chen, Sun Yat-Sen University, China, chenl46@mail.sysu.edu.cn
- Wuling Huang, Chinese Academy of Science, China wuling.huang@ia.ac.cn
- Yier Jin, University of Florida, USA, yier.jin@ece.ufl.edu
- Abhishek Murthy, Philips Lighting Research, USA abhishek.murthy@philips.com
- Rajiv Ranjan, Newcastle University, United Kingdom, raj.ranjan@ncl.ac.uk
- Muhammad Shafique, Vienna University of Technology, Austria, mshafique@ecs.tuwien.ac.at
- Yiyu Shi, University of Notre Dame, USA, yshi4@nd.edu
- Ming-Chang Yang, Chinese University of Hong Kong, Hong Kong, mcyang@cse.cuhk.edu.hk