School of Computer Science & Electronic Engineering

Subject:

Distributed AI for connected IoT devices

Submitted as a part of:

CE902 Professional Practice and Research Methodology

Author:

Karan Bhatt

(Registration: 2112108)

Supervisors:

Dr. Amit Singh

Contents

Abstract

Technologies and Devices that are/can be connected to the internet are growing extensively that lead to the development of the Internet-of-Things, it is efficient to interconnect different devices via wireless technology. It is also known as massive machine type communication (mMTC) [1]. as it generates massive data that data provides valuable information and knowledge about the users. We can use Artificial Intelligence to utilise the data by processing that data to generate beneficial insights and prediction for decision making [2]. In this article we are going to explore distribution of computations from edge devices to cloud so we can achieve the user requirements of privacy, security, processing power and energy efficiency. [3] The reaction speed demanded by such devices involves the processing of IoT data at the edge, yet the edge often lacks the capacity to train Artificial intelligence models. We propose a framework for Artificial Intelligence algorithms that retains the benefits of both edge processing and server / cloud-based computing. [4]. We are going to use deep learning methods DNN/CNN to analyse our data because they can be used to analyses the big data as it has higher accuracy rate than any other techniques. We can deploy such models to our cloud- based computer side as it can analyse the data and give the accurate results in mean time, Because of its processing power efficiency.

Introduction:

Internet-of-Things (IoT) technology has been extremely beneficial in terms of increasing automation, efficiency, and consumer comfort. As it can be used in many fields like Health care, smart cities, self-driven cars etc. The number of IoT devices has exploded. It is predicted to surpass 8.4 billion devices in 2020 and hit 20.4 billion devices in 2022, According to Gartner, we will have more than 30 billion IoT devices in 2025 [5]. The current improvements in generation and the fast merger of regions, inclusive of observing and operating technology, implant systems, wi-fi conveying, and statistics analytics, are rushing up the improvement of the Internet of Things. The IoT is used for accumulating massive enterprise dividends. The impulsive enlargement withinside the variety of gadgets connected to the IoT and the exponential surge in statistics usage demonstrates how the boom of massive statistics and IoT coincide with every other. The composition of those fields is touching nearly all regions of technology and businesses. That would be resulting in massive traffic and data exchange between data providers and customers [3]. In order to obtain information from massive amounts of IoT data in real time, processing must take place close to the data is created (the edge). However, most of the machine learning, Deep learning and Artificial Intelligence algorithms necessitate a substantial amount of system resources, which may not always be accessible at the edge [4]. So, we need to do that with the help of server / cloud-based computing. Environmental context analysis is essential in IoT applications, but it is also one of the most difficult tasks due to the usually massively dispersed, diversified, complicated, and noisy sensing scenarios. [6] Deep/Convolutional Neural Networks (DNNs/CNNs) have undergone tremendous research and are frequently utilized in big data processing because of its accuracy rate is equivalent to human experts [7]. That's why analysis techniques and algorithms based on DNN / CNN techniques became the solution to deal with the big data generated by the IoT devices. Local DNN inference on smartphones and embedded devices, on the other hand, necessitates high computation power and memory footprints [3], which are typically not accessible in IoT computing systems. Cloud-assisted techniques trade-off localized Deep Neural Network inferences performance for sometimes

unexpected distant server availability and network connection delay, as well as possibly revealing confidential and sensitive data throughout data transmission and remote processing. There are numerous challenges and limits to cloud-based processing of consumer data. Exporting data towards the cloud, for example, it creates privacy and security issues. Due to increasing awareness of the privacy and security issues of internet platforms, and they are expected to become more cautious about gadgets that send sound - visual information to the web for additional processing. Laptop cameras, in-home cameras, smart speakers, and so on are examples. Additionally, several IoT devices need quick decision making, that makes cloud-based computing unfeasible because it increases communication latency. [9] To address the above - mentioned challenges, several attempts were undertaken to push machine learning inference from the cloud and onto the edge. Edge processing provides the advantage of maintaining data nearer to its origin, allowing for real-time answers while maintaining end-user privacy [10, 11]. However, edge computing with machine learning techniques has its own set of issues. Since most machine learning algorithms seem to be computationally costly, edge devices are insufficient for such workloads because of their limited performance speed, power, and storage capabilities. To that purpose, a substantial amount of research work has been done to examine efficient techniques for deploying Deep Neural Networks to the edge. I propose a structure based on a distributed edge/cloud methodology that offers a reasonable compromise between the advantages and disadvantages of data processing at the edge vs centrally (server / cloud).

Related Work:

Several methodologies to integrating deep learning frameworks into IoT applications have been developed. DNN inference is largely transferred to cloud servers [12, 13] to reduce computation time and edge device resource utilization. In such cloud-assisted methods, however, inference performance is dependent upon unpredictability of cloud availability and connection delay. But that can create privacy issue. Cloud based systems are on layer-based partitioning. Various statements and reduction approaches have been developed to directly install Deep Neural Networks on resource-constrained edge devices [14], [15]. Compression methods have been developed recently to simplify neural network topologies and thereby lower complexity [16]-[18]. Off-loading strategies for cloud computing give a distinct perspective. The objective is not to lower the intricacy of Deep Learning solutions, but to transfer computationally intensive process to high-performing distributed units. For example, [19] suggests a model for efficiently offloading code in an ubiquitous system composed of mobile nodes by either reducing total communication delay or overall energy utilization. [20] suggested a high-level programming language for developing apps for Fog-Computing Sensor Networks that may conceal the diversity of multiple processors and their location in space.

The work in [21] proposes deploying Deep Neural Networks on the collection of smartphone devices inside a wireless local area network (WLAN) in the context of mobile computing. To simultaneously organise the Convolutional Neural Network inference computations between a fixed number of mobile devices. Individual layers are partitioned into slices to boost efficiency and decrease memory footprint, but slices are still performed layer-by-layer in a centrally controlled, bulk-synchronous, and lock-step method. We, on the other hand, fuse layers and employ tighter grid partitioning to reduce

communication, synchronisation, and memory overhead. CNN accelerators have previously used layer fusion methods [22].

Cloud- Edge Architecture:

In a wide area network context, such as the Internet, cloud-edge architecture can be implemented over many network domains. Some big IoT data must be analysed at the edge, but a large portion must be sent to the cloud for deep analysis [24]. For many processes, including configuration and optimization, cloud outcomes may be pushed down to the edge and even IoT [24]. The North-South connection is involved in this process. The EastWest connection is achieved by distributed edge deployment across several domains. In the context of 5G mobile networks, Taleb et al. [23] proposed an architecture for offering video content delivery network capabilities as a service (called CDN slice) through a cross-domain cloud-edge environment. The authors employed edge computing and network functions virtualisation (NFV) to drive allocation of resources and management for CDN slices across various domains [24].

Privacy and Security:

Another major challenge for cloud-edge connection architecture is security and privacy [25], [26], [27]. The cloud is often administered by a third party, and edge clouds could be owned by various service providers. Various data, including user data, network operation data, and business operation data, may need to be transported to the cloud and the edge for processing in order to accomplish a smart choice to aid in the automation of company operations [24]. Malicious data will lead to wrong judgments, ultimately compromising the operation of IoT applications [24]. Excessive data (or private data) exposure might boost the performance of learning models; however, it may breach data protection regulations/laws [24].

Background:

A Convolutional Neural Network (CNN) is made up of main three layers. (1) input layer, (2) output layer, and (3) several hidden layers. Convolution layers, normalisation layers, ReLU layers (i.e., activation functions), pooling layers, and fully connected layers are common components of the hidden layer [9]. Each layer will create higher level of input data. The Batch normalisation layer normalises information across geographically organized extracted features to help minimize covariate shift. To maximize nonlinearity, an activation layer uses an element-wise activation function towards the input data, such as rectified-linear unit (relu),[8,9] that eliminates negative numbers from a feature map by assigning it to zero. we group both the layers with the prior layer that creates their input. [8,9] A pooling layer executes downsamples all along spatial dimensions. It's being used to decrease the spatial dimension of the representations, the number of parameters, the memory footprint, and the frequency of processing in the network gradually. As a result, it also regulates overfitting [9]. The neural network's high-level processing is carried out via fully connected layer. The Softmax and argmax layers generate a probability distribution across the classes to be classified and choose the one with the highest probability as the predictions. The fully connected and convolution layers of a Convolution Neural Network model are one of the most computed and data-intensive [8, 9].

Methodology:

This article presents a methodology for receiving technical constraints associated with IoT units and Deep learning trained structures and providing the optimal distributed task of the Deep learning computation (layers) to the IoT units by reducing information gathering to decision delay. The methodology was already adapted specifically to Convolutional Neural Networks (CNNs), which represent the state-of-the-art in graphical data processing. Surprisingly, these CNNs may work consecutively on the processing pipeline (all layers are completed until the final classification) [29], [30], [31], or at run-time based on the information content of the input [32], [33]. Both circumstances are taken into account in the suggested approach, which can also be employed with numerous CNNs working on the same network of IoT embedded devices (possibly sharing processing layers).

In terms of the article, the following are the primary novel characteristics of the suggested methodology:

- The technique can account for the IoT units' connection and compute abilities, as well as their memory limits.
- The approach may allow CNNs which operational pipeline is determined by the input picture.
- Multiple CNNs (potentially shared processing layers) may run along the same IoT device network.

The following methodology was tested using simulations as well as a real-world technical situation.

Deep convolutional neutral networks (CNNs) have improved throughout time and become more precise and quicker. Since AlexNet [34], innovative structures such as VGG [35], GoogLeNet [36], ResNet [37, 38], DenseNet [39], ResNeXt [33], SE-Net [9], and autonomous neutral architecture search [39, 18, 21] have greatly increased ImageNet classification accuracy.

Apart from accuracy, computing complexity is a significant factor to consider. Real-world activities frequently strive for maximum accuracy within a constrained computational budget dictated by the target platform (e.g., hardware) and application situations (e.g., auto driving requires low latency). This drives a number of works, including Xception [2], MobileNet [8], MobileNet V2 [24], ShuffleNet [35], and CondenseNet [10], to develop lightweight architectures with better speed-accuracy tradeoffs. These works rely heavily on group convolution and depth-wise convolution.

As shown in the figure first we will gather the data from different IoT devices like mobile phone, Camera, Smart watches etc. then we process the CNN on the data gathered from the different devices. There we will do the AI analysis on the edge server, and if the data is big and it requires more processing power then we need to send the data to the cloud server there it will process the data and will provide us the important information about the data in mean time with low latency. As the process is big and it might risk the data loss or data theft, we can use the encoder and decoder method for the same. So, like that the data would be encoded in the secure format and whenever it reaches to the cloud server it will start decoding the data. Once decoding of data is finished the server will start processing the data in CNN model and then it will send the important data or information back to the edge. Where same encoder and decoder method must apply. To get low delay of this process we must use the 5G technology which has the bandwidth over 1000 Mbit/S. with the 28 GHz and 39GHz frequency. Which is way more than 4G networks. And even we can see the 6G in the nearer future so fasten data transfer with fast bandwidth would be feasible to gain low latency.

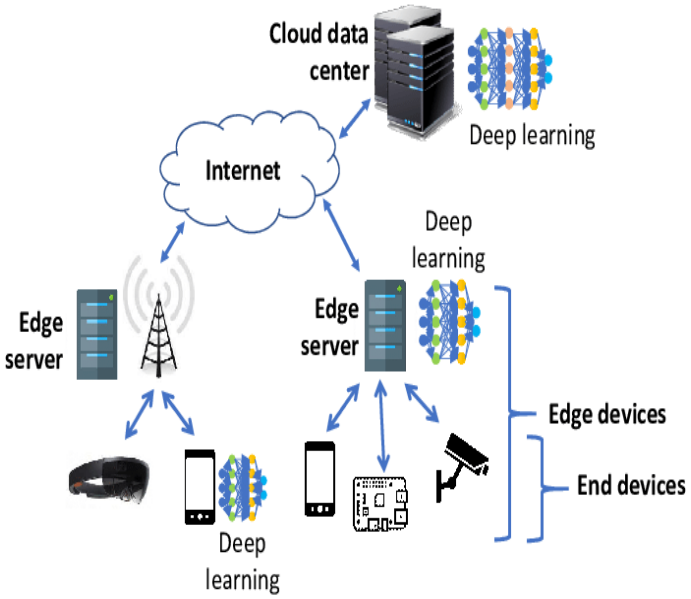


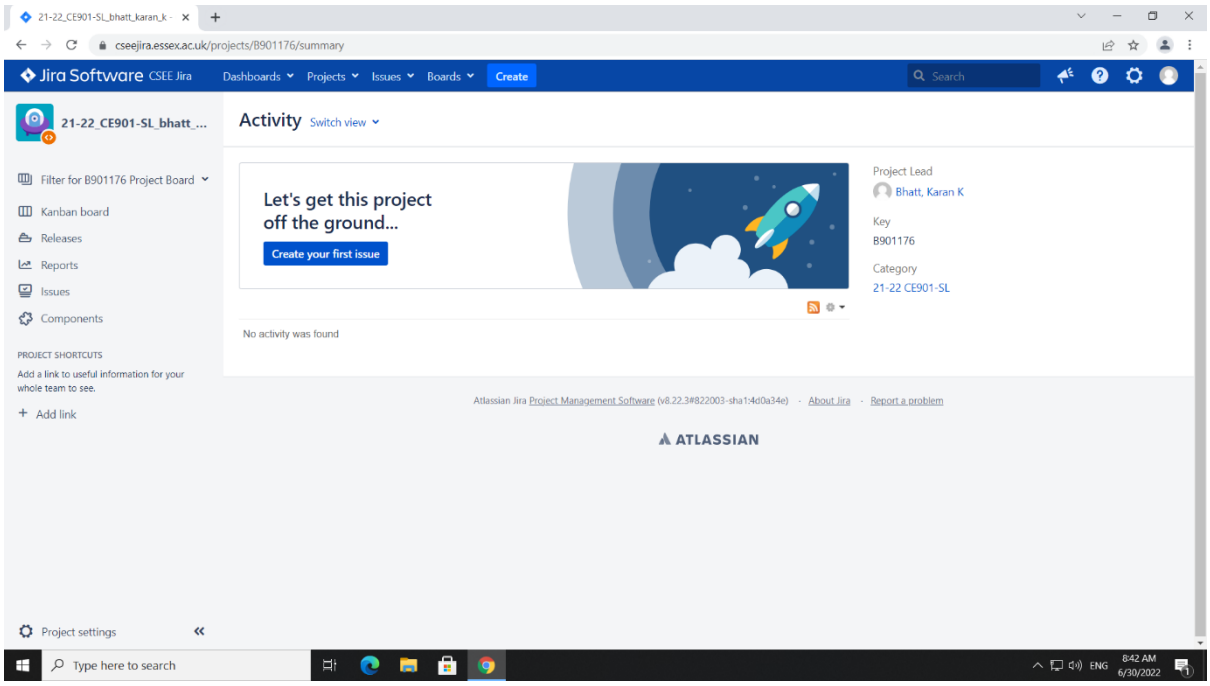
Fig. 1: Deep learning can execute on edge devices (i.e., end devices and edge servers) and on cloud data centres [28] .

Project Planning:

For the Project planning and update the progress we can use these platforms

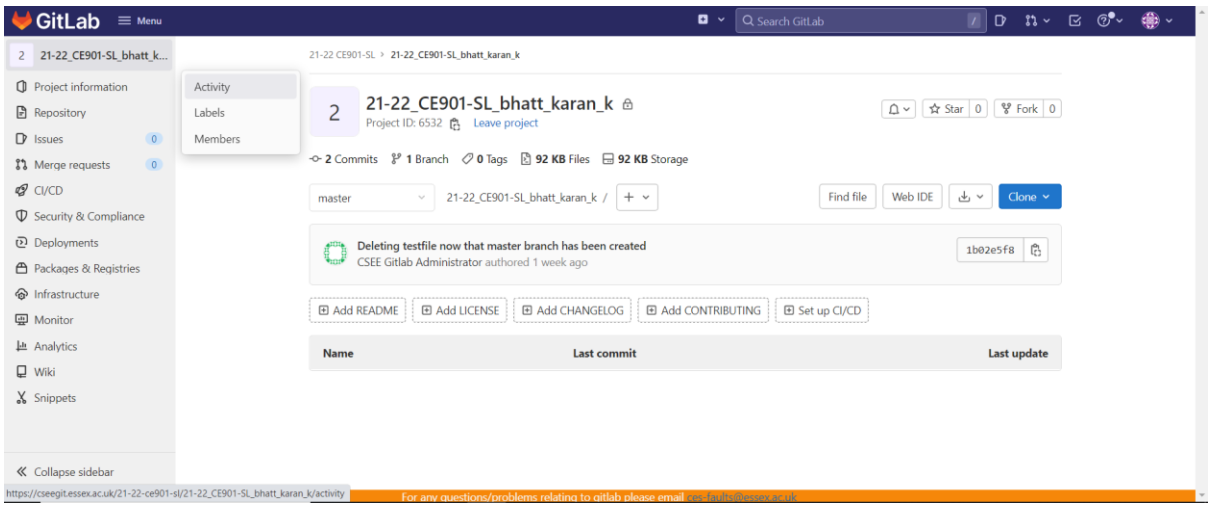
1) JIRA

We can use JIRA for work distribution and complete the project with planning and within the given time.



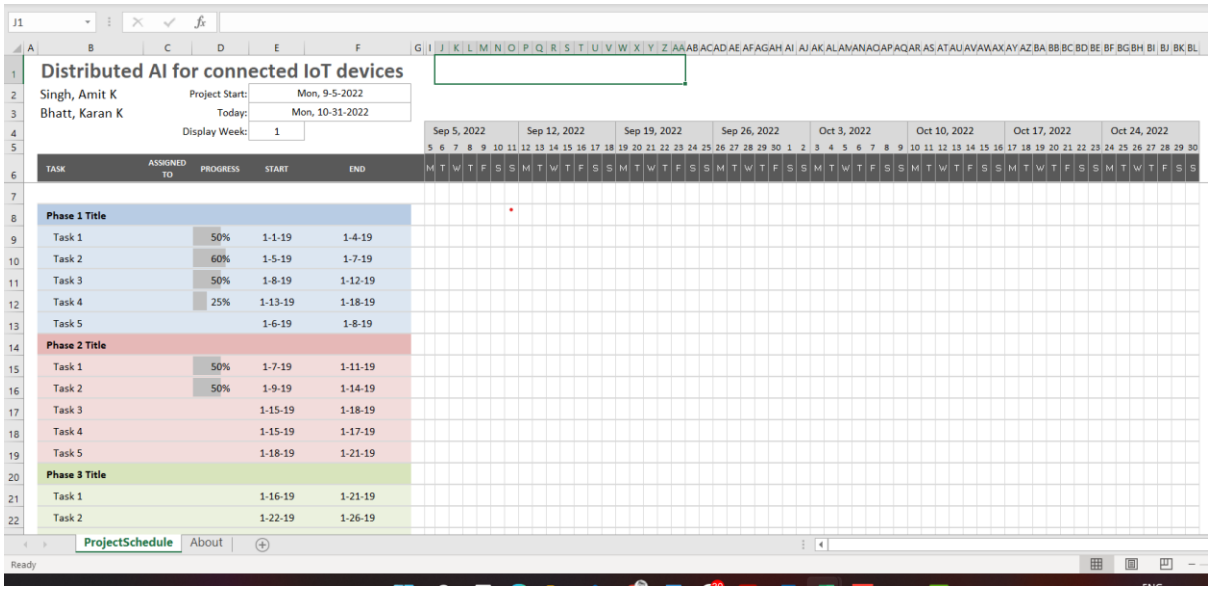
2) GitLab

We can use the GitLab to share the code and check the code if it is right or not or how much coding work is done or how much left.



3) Gantt Chart.

We can use the Gantt chart to keep the record of each and every work we need to do or we have already done, how many tasks are pending and how many are completed or how much it is completed.



Distributed AI for connected IoT devices

Singh, Amit K
Bhatt, Karan K

Project Start: Mon, 9-5-2022
Today: Mon, 10-31-2022
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Title				
Task 1		50%	1-1-19	1-4-19
Task 2		60%	1-5-19	1-7-19
Task 3		50%	1-8-19	1-12-19
Task 4		25%	1-13-19	1-18-19
Task 5			1-6-19	1-8-19
Phase 2 Title				
Task 1		50%	1-7-19	1-11-19
Task 2		50%	1-9-19	1-14-19
Task 3			1-15-19	1-18-19
Task 4			1-15-19	1-17-19
Task 5			1-18-19	1-21-19
Phase 3 Title				
Task 1			1-16-19	1-21-19
Task 2			1-22-19	1-26-19

Sep 5, 2022 | Sep 12, 2022 | Sep 19, 2022 | Sep 26, 2022 | Oct 3, 2022 | Oct 10, 2022 | Oct 17, 2022 | Oct 24, 2022

ProjectSchedule | About

Ready

References:

[1] H. Song, J. Bai, Y. Yi, J. Wu and L. Liu, "Artificial Intelligence Enabled Internet of Things: Network Architecture and Spectrum Access," in IEEE Computational Intelligence Magazine, vol. 15, no. 1, pp. 44-51, Feb. 2020, doi: 10.1109/MCI.2019.2954643.

[2] S. Alrubei, E. Ball and J. Rigelsford, "The Use of Blockchain to Support Distributed AI Implementation in IoT Systems," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3064176.

[3] N. Dhieb, H. Ghazzai, H. Besbes and Y. Massoud, "Scalable and Secure Architecture for Distributed IoT Systems," 2020 IEEE Technology & Engineering Management Conference (TEMSCON), 2020, pp. 1-6, doi: 10.1109/TEMSCON47658.2020.9140108.

[4] S. B. Calo, M. Touna, D. C. Verma and A. Cullen, "Edge computing architecture for applying AI to IoT," 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 3012-3016, doi: 10.1109/BigData.2017.8258272.

[5] "Gartner says the internet of things will transform the data center," Gartner Inc., 2014. [Online]. Available: http://www.gartner.com/newsroom/id/2684616

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[8] Z. Zhao, K. M. Barijough and A. Gerstlauer, "DeepThings: Distributed Adaptive Deep Learning Inference on Resource-Constrained IoT Edge Clusters," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2348-2359, Nov. 2018, doi: 10.1109/TCAD.2018.2858384.

[9] Li Zhou, Mohammad Hossein Samavatian, Anys Bacha, Saikat Majumdar, and Radu Teodorescu. 2019. Adaptive parallel execution of deep neural networks on heterogeneous edge devices. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing (SEC '19). Association for Computing Machinery, New York, NY, USA, 195–208. https://doi.org/10.1145/3318216.3363312

[10] Ramyad Hadidi, Jiashen Cao, Micheal S Ryoo, and Hyesoon Kim. 2019. Collaborative Execution of Deep Neural Networks on Internet of Things Devices. arXiv preprint arXiv:1901.02537 (2019).

[11] He Li, Kaoru Ota, and Mianxiong Dong. 2018. Learning IoT in edge: deep learning for the internet of things with edge computing. IEEE Network 32, 1 (2018), 96–101.

[12] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, 2017, pp. 615–629.

[13] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 328–339.

[14] S. Bhattacharya and N. D. Lane, "Sparsification and separation of deep learning layers for constrained resource inference on wearables," in *Proc. ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2016, pp. 176–189.

[15] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, "DeepIoT: Compressing deep neural network structures for sensing systems with a compressor-critic framework," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2017, pp. 1–14.

[16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv preprint arXiv:1602.07360*, 2016

[17] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," *arXiv preprint arXiv:1707.01083*, 2017

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ser. Mobi- Hoc '12. New York, NY, USA: ACM, 2012, pp. 145–154.

[20] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, ser. MCC '13. New York, NY, USA: ACM, 2013, pp. 15–20.

[21] J. Mao, X. Chen, K. W. Nixon, C. Krieger, and Y. Chen, "MoDNN: Local distributed mobile computing system for deep neural network," in Proc. Design Autom. Test Europe Conf. Exhibit. (DATE), 2017, pp. 1396–1401.

[22] M. Alwani, H. Chen, M. Ferdman, and P. Milder, "Fused-layer CNN accelerators," in Proc. Int. Symp. Microarchit. (MICRO), 2016, pp. 1–12.

[23] T. Taleb, P. A. Frangoudis, I. Benkacem, and A. Ksentini, "Cdn slicing over a multi-domain edge cloud," IEEE Transactions on Mobile Computing, pp. 1–1, 2019.

[24] Y. Wu, "Cloud-Edge Orchestration for the Internet of Things: Architecture and AI-Powered Data Processing," in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12792-12805, 15 Aug.15, 2021, doi: 10.1109/JIOT.2020.3014845.

[25] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu, "Privacypreserving online task allocation in edge-computing-enabled massive crowdsensing," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7773–7787, 2019

[26]] P. Zhou, K. Wang, J. Xu, and D. Wu, "Differentially-private and trustworthy online social multimedia big data retrieval in edge computing," IEEE Transactions on Multimedia, vol. 21, no. 3, pp. 539–554, 2019

[27]] N. Zhang, R. Wu, S. Yuan, C. Yuan, and D. Chen, "Rav: Relay aided vectorized secure transmission in physical layer security for internet of things under active attacks," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8496–8506, 2019.

[28] Deep Learning With Edge Computing: A Review - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Deep-learning-can-execute-on-edge-devices-ie-end-devices-and-edge-servers-and-on_fig1_334489669 [accessed 30 Jun, 2022]

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS '12, vol. 1. Curran Associates Inc., 2012, pp. 1097–1105.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), ser. CPVR '16. IEEE, jun 2016, pp. 779–788

[32] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama, “Adaptive neural networks for efficient inference,” in Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 527–536.

[33] S. Disabato and M. Roveri, “Reducing the computation load of convolutional neural networks through gate classification,” in 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018, pp. 1–8.

[34] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[35] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[36] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions. Cvpr (2015)

[37] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[38] He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European Conference on Computer Vision. pp. 630–645. Springer (2016)

[39] Huang, G., Liu, Z., Weinberger, K.Q., van der Maaten, L.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. vol. 1, p. 3 (2017)