A Low Power And High Performance Artificial Intelligence Approach To Increase Guidance Navigation And Control Robustness

Pablo Ghiglino
Klepsydra Technologies
pablo.ghiglino@klepsydra.com

Mandar Harshe
Klepsydra Technologies
mandar.harshe@klepsydra.com

ABSTRACT

New generations of spacecrafts are required to perform tasks with an increased level of autonomy. Space exploration, rendezvous services, space robotics, etc. are all growing fields in Space that require more sensors and more computational power to perform these missions. Furthermore, new sensors in the market produce better quality data at higher rates while new processors can increase substantially the computational power. Therefore, near-future spacecrafts will be equipped with large number of sensors that will produce data at rates that has not been seen before in space, while at the same time, data processing power will be significantly increased. In regards to guidance navigation and control applications, vision-based navigation has become increasingly important in a variety of space applications for enhancing autonomy and dependability. Future missions such as Active Debris Removal will rely on novel high-performance avionics to support image processing and Artificial Intelligence algorithms with large workloads. Even more complex is the case of vision-based precision landing, that high rate processing is a must and can be the tipping point of a successful mission. This new scenario of advanced Space applications and increase in data amount and processing power, has brought new challenges with it: low determinism, excessive power needs, data losses and large response latency. In this article, a novel approach to on-board artificial intelligence (AI) is presented that is based on state-of-the-art algorithmic trading software techniques, which is a field that underwent a similar challenge, although is a different scale, in the early 2010. The approach presented here optimizes the limited available computing resources, and makes AI applications much more reliable, therefore somewhat reshaping the paradigm of embedded software engineering. A benchmarks is presented here for a pose estimation of the asteroid 67P/Churyumov–Gerasimenko using AI base of images from the Rosetta mission. In this paper, we show that the data processing rate and power saving of the applications increase substantially with respect to standard AI solutions.

Introduction

It is common ground that embedded systems have evolved hugely in the last decade.[1] New generations of autonomous embedded systems are required to perform more and faster on-board data processing. Sensors, embedded processors, and hardware in general have hugely evolved in the last decade, equipping embedded systems with large number of sensors that will produce data at rates that has not been seen before while simultaneously having computing power capable of large data processing,[2].[3] However, embedded software engineering has remain virtually unchanged for the last two decades, making the development of advanced application extremely cumbersome, error-prone, sub-optimal and usually delay incurring.[4] Although the work presented here is general to all embedded systems, the specific fields where tests were performed are AI onboard and a real space application. Hence, the specific discussion about the state of the art in both fields is included in this introduction.

AI Space systems

Space applications requiring on-board signal processing at high rate is a growing field. It is particularly important the use of AI for reducing signal-to-noise-ratio (SNR). Typical applications are Internet of Things (IoT), traffic control, telecomms application, etc.

Another field where AI is becoming critical is Earth Observation. There is a growing number of Synthetic Aperture Radar (SAR) sensors in satellites due to the increasing demand for Earth applications for SAR data. The amount of data produced by a SAR sensor prevents real-time data transfer to the ground due to the limitations of downlink speeds, thus requiring large on-board data storage. Several high-level solutions have been proposed to improve this:

- Use specialised on-board compression algorithms.
- Use on-board Artificial Intelligence (AI) to filter irrelevant or low quality data and send only a subset of data.

Space autonomous navigation systems

In a different field, vision based navigation, there is also a challenge of data processing combined with AI algorithms. One example is rendezvous with uncooperative objects in space, e.g., debris removal,[5],[6].[7] Another example of this is autonomous pinpoint planetary landing, where the number of sensors and the complexity of the Guidance Navigation and Control (GNC) algorithms make this discipline still one of the biggest challenges in space,[8],[9].[10] One common element to these two use cases, is a well known fact in control engineering: for optimal control algorithms, the higher the rate of sensor data, the better is the performance of the algorithm.[11]

Inference in Artificial Intelligence

There are several components to artificial intelligence (1). First, there is the training and design of the model. This activity is usually carried out by data scientists for a specific field of interest. Once the model is designed and trained, the model is deployed to the target computer for real-time execution. This is what is called inference. Inference consists of two parts, the trained model and the AI inference engine to execute the model. The focus of this research has been solely on the inference engine software algorithms.

Trends in Artificial Intelligence inference acceleration

The most common operation in AI inference by far is matrix multiplications. These operations are constantly repeated for each input data to the AI model. In recent years, there has been a substantial development in this area with both industry and academia progressing substantially in this field. While the current trend is to focus on hardware acceleration like Graphic Processing Units (GPU)[12] and Field-programmable gate array (FGPA),[13] these techniques are currently not broadly available to the Space industry due to radiation issues and excessive energy consumption for the former, and programming costs for the latter. The use of CPU for inference, however, has been also undergoing an important revolution when the CPU core has a Floating processing unit (FPU) connected to it.[14] CPUs are widely used in Space due to large Space heritage and also ease of programming and use. Several AI inferences engines are available for CPU+FPU setups. The work presented here will show the results of extensive research in building a new AI inference that both reduces power consumption and also increases data throughput.

Parallel processing applied to Artificial Intelligence inference

Within the field of inference engines for CPU+FPU, there has been an over-focus on matrix multiplication parallelisation,[15].[16] This process consists in splitting the operations required for a matrix multiplication into smaller to be executed by several threads in parallel. Figure 2 shows an example of this type of process, where rows from the left-hand matrix and columns from the right-hand matrix are individual operations to be executed by different threads.

The theoretical advantage of this approach is its minimal latency.[15] However, there is an emerging alternative approach to parallelisation, which is based in the concept of pipeline.[17] This approach works in a similar manner to an assembly line, where each part of this line corresponds to a complete matrix multiplication. This approach is particularly well suited for AI deep neural networks (DNN). Figure 3 shows this approach. The main advantage make pipeline a reliable approach to data processing in resources constraint enviroment, like Space on-board computers, is its higher throughput: pipelining can enable a substantial increase in throughput with respect to traditional parallelisation.[18]

Pipelineing of a matrix multiplication sequence

Combining the concept of pipelining above with lock-free algorithms,[19] the authors have developed

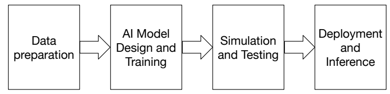

Figure 1: AI main components

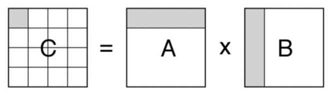

Figure 2: Parallel matrix multiplication

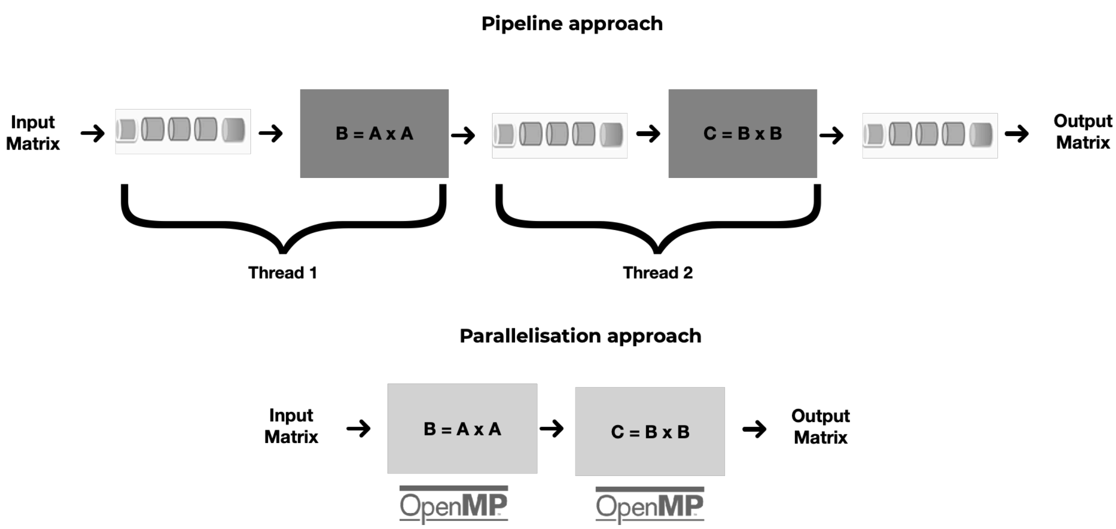

Figure 3: Pipelining vs parallelisation

a new pipelining approach that can process data at 2 to 8 times increased data rate, while at the same time reduce power consumption up to 75%. This new pipelining algorithm consists of thee main elements:

- Use of lock-free eventloops to connect the matrix multiplications operations.
- Use of FPU vectorisation to accelerate the matrix multiplications
- One eventloop per thread, meaning that each matrix multiplication happens in one thread.

This novel approach can be seen in figure 4.

Matrix multiplication benchmark

In order to validate the previous approach, a basic benchmark was performed with the following setup:

- A sequence of n matrix multiplications, With $n = [10 - 60]$
- An increase rate of data from 2Hz to 100Hz.
- Each matrix is a squared matrix of 100 side of float numbers.
- The proposed solution was compared with OpenMP approach as suggested in figure 3

The test were perform in an AMD64 4-core computer with the results presented in images 5 - 12. In conclusion, these results show that the new proposed pipelining approach is extremely efficient, providing up to 4 times increase in data processing, with up to 75% reduction in CPU usage.

AI inference pipelining

In the section we present the main aspect of the author novelty, which is the application of the above presented lock-free pipelining. This is achieved by applying the pipelining approach where each layer is considered an individual operation, as shown in figure 13

There results of this pipelining can be seen in the next section.

Experimental Setup

The experimental setup consists of testing of an AI model for pose estimation of the asteroid 67P/Churyumov–Gerasimenko using AI base of images from the Rosetta mission. The benchmark was done comparing the propose AI inference with respect to the two main market leaders TensorFlowLite and OpenCV-CNN. The test was done in the two different computers with the following parameters:

- CPUs: ARM64 (2-cores, x86_64 (4-cores)
- Data rate: 5 FPS for ARM64, 10FPS for x86_64
- Performance criteria: throughput, latency, CPU usage and RAM

The results are shown in the tables 1 and 2.

Results analysis

Table 1: AI Benchmarks for x86_64

Criteria	Pipeline AI	TFL	OCV-CNN
Throughput	10	4.8	3.6
Latency	8ms	11ms	25ms
CPU	24	51	32
RAM	450Mb	425Mb	375Mb

Table 2: AI Benchmarks for ARM64

Criteria	Pipeline AI	TFL	OCV-CNN
Throughput	5	2.6	2.2
Latency	15ms	18ms	32ms
CPU	48	76	56
RAM	450Mb	415Mb	354Mb

This results show ground breaking performance by duplicating the data rate and reducing power consumption by 50%. All these is achieved with keeping RAM and latency not only within acceptable limits, but also improving latency.

These results varied from model to model, but with an overall improvement when using the pipelining AI algorithm.

Conclusions

In this paper, we have presented the state of the AI techniques for on-board computers for Spacecrafts. We have presented the two main approaches to data processing acceleration, i.e., FPGA/GPU and CPUs. Within the CPU approach, we have covered the different approaches to data parallelisation currently in the market, including the here presented pipelining approach. This paper has shown that lock-free pipelining is extremely efficient for matrix multiplications and specifically to AI applications.

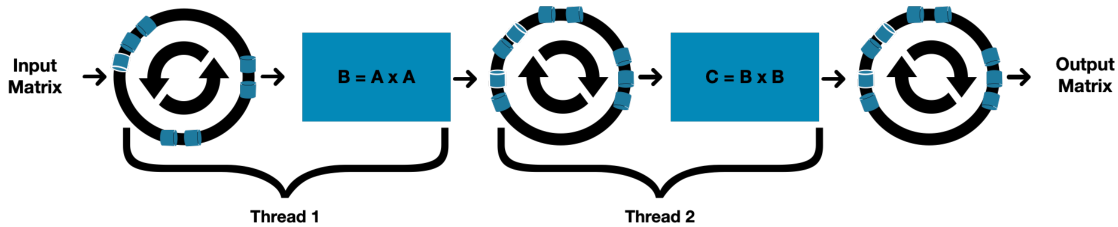

Figure 4: Novel proposed pipelining approach

This techniques can both substantially increase data throughput, reduce power consumption, while at the same time keeping the latency, and RAM completely under control and event with improvement with respect of the industry standard tools.

Lock-free pipelining advantages are particularly beneficial to Space applications, in particular for planetary landing, where the rate of data is quite high and required very high responsiveness. Similarly, for Earth Observation application, lock-free pipelining is very well suited for the current large volume of data requirements.

Future Work

In terms of future work, the main area of research are two. First, is the expansion of the current pipelining to support not only sequential DNNs but also graph DNNs that require a different configuration and approach to pipelining. Preliminary tests show extremely promising results with even higher performance gains.

Secondly, the validation of lock-free pipelining in real-time operating systems is also an area of large research. While a substantial amount of research has to be carried out still, preliminary results of matrix multiplication are quite promising.

References

[1] P. Sharma, H. Verma, V. Negi, A. Sharma, S. Banarwal, and G. Verma. Evolutionary trends in embedded system design. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 2059–2062, March 2016.

[2] F. Samie, L. Bauer, and J. Henkel. Iot technologies for embedded computing: A survey. In *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 1–10, Oct 2016.

[3] A. Banerjee, A. Mondal, A. Sarkar, and S. Biswas. Real-time embedded systems analysis — from theory to practice. In *2015 19th International Symposium on VLSI Design and Test*, pages 1–2, June 2015.

[4] M. V. Woodward and P. J. Mosterman. Challenges for embedded software development. In *2007 50th Midwest Symposium on Circuits and Systems*, pages 630–633, Aug 2007.

[5] O. Kechagias-Stamatis, N. Aouf, and M.A. Richardson. High-speed multi-dimensional relative navigation for uncooperative space objects. *Acta Astronautica*, 160:388 – 400, 2019.

[6] Mah Zarei and Seyed Malaek. Motion estimation of uncooperative space objects: A case of multi-platform fusion. *Advances in Space Research*, 08 2018.

[7] Ksenia Klionovska, Jacopo Ventura, Heike Benninghoff, and Felix Huber. Close range tracking of an uncooperative target in a sequence of photonic mixer device (pmd) images. *Robotics*, 7(1), 2018.

[8] Pingyuan Cui, Xizhen Gao, Shengying Zhu, and Wei Shao. Visual navigation using edge curve matching for pinpoint planetary landing. *Acta Astronautica*, 146:171 – 180, 2018.

[9] Thomas Voirin, Jeff Delaune, Guy Le Besnerais, Jean-Loup Farges, ClÃ©ment Bourdarias, and Hans KrÃŒger. Challenges of pinpoint landing for planetary exploration : The lion absolute vision-based navigation system step-wise validation approach. In *Conference: International Planetary Probes Workshop 10*, 06 2013.

[10] Shuang Li, Pingyuan Cui, and Hutao Cui. Vision-aided inertial navigation for pinpoint planetary landing. *Aerospace Science and Technology*, 11(6):499 – 506, 2007.

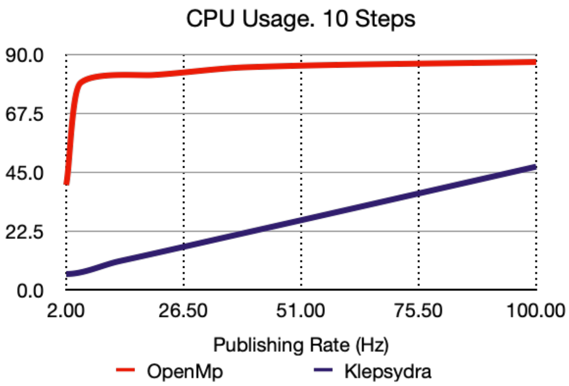

Figure 5: CPU for 10 multiplication steps.

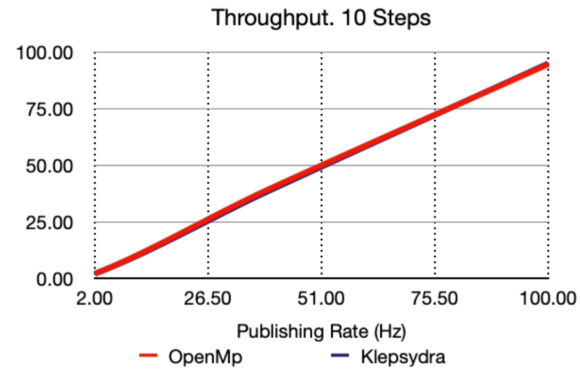

Figure 6: Data throughput for 10 steps.

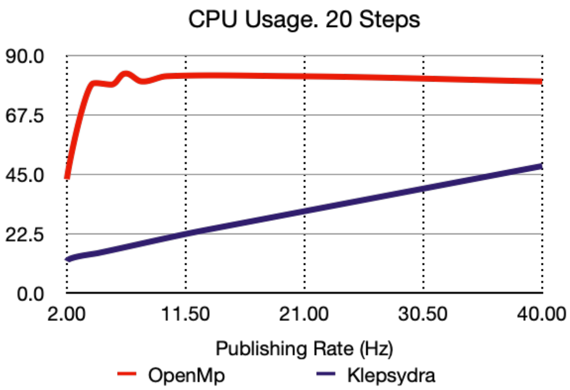

Figure 7: CPU for 20 multiplication steps.

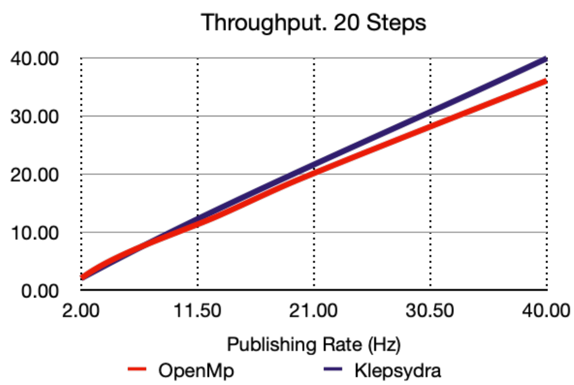

Figure 8: Data throughput for 20 steps.

[11] P. Ghiglino, J. L. Forshaw, and V. J. Lappas. Oqtal: Optimal quaternion tracking using attitude error linearization. *IEEE Transactions on Aerospace and Electronic Systems*, 51(4):2715–2731, Oct 2015.

[12] Toru Baji. Evolution of the gpu device widely used in ai and massive parallel processing. In *2018 IEEE 2nd Electron Devices Technology and Manufacturing Conference (EDTM)*, pages 7–9, 2018.

[13] Ahmad Shawahna, Sadiq M. Sait, and Aiman El-Maleh. Fpga-based accelerators of deep learning networks for learning and classification: A review. *IEEE Access*, 7:7823–7859, 2019.

[14] Haidong Lan, Jintao Meng, Christian Hundt, Bertil Schmidt, Minwen Deng, Xiaoning Wang, Weiguo Liu, Yu Qiao, and Shengzhong Feng. Feathercnn: Fast inference computation with tensorgemm on arm architectures. *IEEE Transactions on Parallel and Distributed Systems*, 31(3):580–594, 2020.

[15] L. Dagum and R. Menon. Openmp: an industry standard api for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55, 1998.

[16] Sunil Shukla, Bruce Fleischer, Matthew Ziegler, Joel Silberman, Jinwook Oh, Vijayalakshmi Srinivasan, Jungwook Choi, Silvia Mueller, Ankur Agrawal, Tina Babinsky, Nianzheng Cao, Chia-Yu Chen, Pierce Chuang, Thomas Fox, George Gristede, Michael Guillorn, Howard Haynie, Michael Klaiber, Dongsoo Lee, Shih-Hsien Lo, Gary Maier, Michael Scheuermann, Swagath Venkataramani, Christos Vezyrtzis, Naigang Wang, Fanchieh Yee, Ching Zhou, Pong-Fei Lu, Brian Curran, Leland Chang, and Kailash Gopalakrishnan. A scalable multi-teraops core for ai training and inference. *IEEE Solid-State Circuits Letters*, 1(12):217–220, 2018.

[17] Geoffrey Biggs, Noriaki Ando, and Tetsuo Kotoku. Rapid data processing pipeline development using openrtm-aist. In *2011 IEEE/SICE International Symposium on System Integration (SII)*, pages 312–317, 2011.

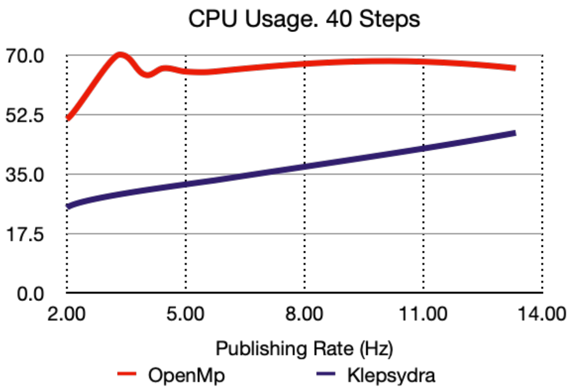

Figure 9: CPU for 40 multiplication steps.

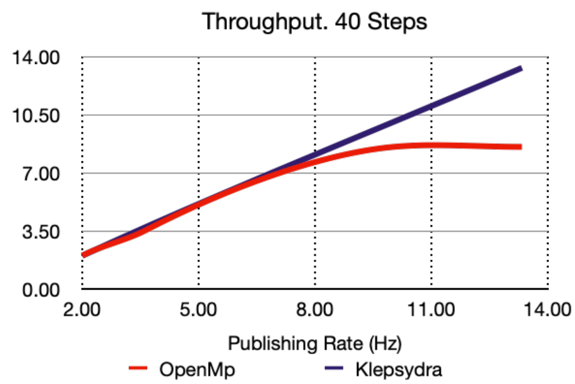

Figure 10: Data throughput for 40 steps.

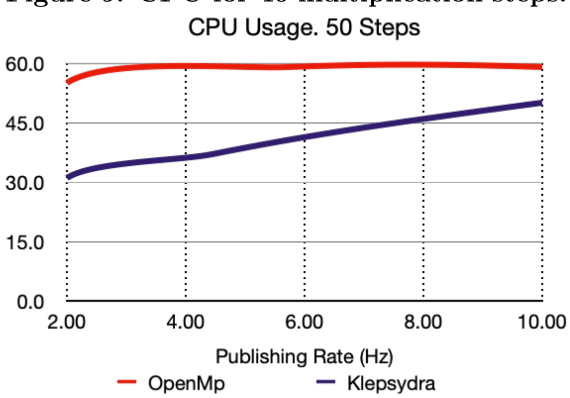

Figure 11: CPU for 50 multiplication steps.

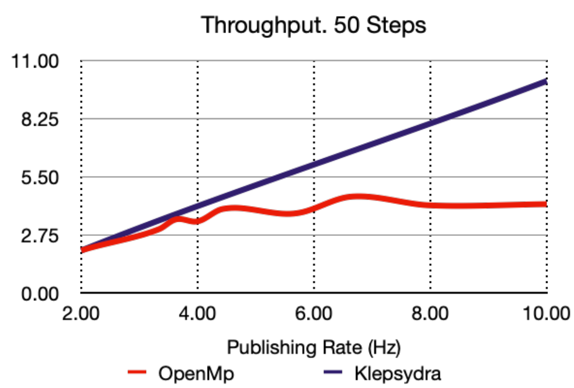

Figure 12: Data throughput for 50 steps.

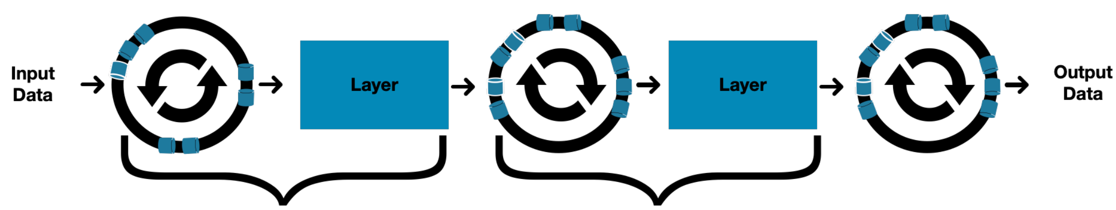

Figure 13: Novel AI pipelining approach

[18] K. Sravani and Rathnamala Rao. High throughput and high capacity asynchronous pipeline using hybrid logic. In *2017 International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*, pages 11–15, 2017.

[19] P. Ghiglino and M. Harshe. A deterministic and high performance parallel data processing approach to increase guidance navigation and control robustness. In *2020 IAF SPACE SYSTEMS SYMPOSIUM (IAC)*, 2020.