OFFLOADING UAV COMPUTATIONS TO THE EDGE

by

Kundan Chaudhary

Submitted in partial fulfillment of the

requirements for Departmental Honors in

the Department of Computer Science

Texas Christian University

Fort Worth, Texas

May 2, 2022

OFFLOADING UAV COMPUTATIONS

TO THE EDGE

Project Approved:

Supervising Professor: Krishna P Kadiyala, PhD

Department of Computer Science

Bingyang Wei, PhD

Department of Computer Science

Gehendra Kharel, PhD

Department of Environmental Science

ABSTRACT

Unmanned Aerial Vehicles (UAVs) navigate by observing their surroundings using on-board cameras and sensors. UAVs also collect data using their onboard devices, compute the data, and analyze the information to decide which direction to move next. Onboard computation and processing are known to have a significant overhead on UAVs and might limit the time of flight, speed and efficiency of UAVs. Thus, the idea of offloading computation and processing to servers that are available at the edge of the network and close to the UAVs has been strongly advocated and researched recently. This paper aims to leverage offloading UAV computations to the edge while exploring different implementations and approaches of computation offloading. To achieve the aim, we reviewed the papers from IEEE conferences and other verified institutions such as NASA and Cornell University and did a literature survey based on our findings. Our results implicate the overview of computation offloading and its implementations for various UAV applications. This study emphasizes on various offloading approaches for UAV computations to Mobile Edge Computing (MEC) servers along with their requirements, advantages, and disadvantages.

Table of Contents

Introduction

Unmanned Aerial Vehicles (UAVs), also referred to as drones, are remotely piloted aircrafts that fly and perform their missions without direct human interference. The UAV system is a low-cost system that works on the basis of photogrammetry and remote sensing. Increasingly in recent years, UAVs have been used in a broad array of applications, ranging from rescue and military operations to agricultural activities such as paddy monitoring or spraying, and other civilian applications (Samad et al., 2013). Figure 1 and figure 2 show the use of UAVs in agriculture for spraying pesticides or capturing detail views of crops, as mentioned in Hayden (2021).

Figure 1: Drones taking an image of crops (Hayden, 2021)

More recently, the use of drones in search and rescue operations has seen practical implementation while also attracting interesting research in the form of case studies, such as those presented in McRae et al. (2019) and Tuśnio & Wróblewski (2021). In McRae et al. (2019) and Tuśnio & Wróblewski (2021) consumer drones were used by first responders to perform operations in mountains to rescue mountaineers. According to Chadha (2020), “in 2017 alone, the United States National Park Service deployed almost 3,500 search and rescue missions in national parks”.

Figure 2: Drones spraying pesticide over crops (Hayden, 2021)

According to Intelligence (2022), it is predicted that the total global shipment of UAVs would reach 2.4 million in 2023, which means it would be increasing at a 66.8% compund annual growth rate. As the applications of UAVs are increasing day by day, UAVs are also expected to be more efficient, guarantee autonomy and safety while successfully completing their missions under time and flight constraints (Chen et al., 2021). This means that the UAVs need to have better awareness of the environment they are deployed in and also have decision-making ability. To achieve high efficiency, UAVs commonly analyze the input from its on-board camera and other sensors, making decisions and acting to ensure mission success. This loop of capturing and analyzing real-time image/video input and making decisions during the flight is critically dependent on the computational and analytical abilities available to the UAV.

The technology used in UAV might vary slightly depending on their application, but it is imperative that every UAV has some common components that the battery life needs to support. The primary components required for UAV aviation are standard propellers, pusher propellers, brushless motors, motor mounts, landing gear and boom (*A drone's components: Guide for beginners,* 2021). Additionally, to keep steady communication and flight control, the UAV must

also have an electronic speed controller, flight controller, GPS module, receiver, and antenna (*A drone's components: Guide for beginners,* 2021). Apart from these general components, depending on the application requirements, components such as camera, 3d sensors, collision avoidance sensors, and gimbal may also be needed (*A drone's components: Guide for beginners,* 2021). Please refer to the article, a drone's components: *Guide for beginners for beginners* (2021) to learn more about the components. In figure 3 below, we see the many components of a DJI Phantom4 drone *Anatomy of a drone infographic* – this is purely for the purpose of reviewing many components of a drone together, but we do not use this drone in any specific implementations in the work presented here.

Figure 3: Inside a Phantom DJI4 drone (Anatomy of a drone infographic, 2022)

It is obvious that if the battery on-board on the UAV has to support the computational and analytical abilities alongside the primary hardware components, it would decrease the UAV's efficiency and longevity. To avoid this, researchers, today, are looking in to computation offloading, which simply means to "offload" computation to a remote server, instead of doing it on board. The goal of this paper is to explore computation offloading, comparing different approaches to implementations of offloading UAV computations to the edge and presenting one specific implementation. This paper is organized in six sections as follows. In the first section, we will present the problem in-depth and in the second section, we will explain the methodology for this research. Following that, in the third section, we will learn about computation offloading and edge computing. In the fourth section, we will discuss one of the existing edge computing implementations called DeepBrain that we find is one of the most applicable today. In the fifth section, we present with a result and discussion section where we provide summary of three other implementations and compare it with DeepBrain. Finally, we conclude the paper in the sixth section with the summary and overview of the paper.

Problem Motivation

The primary objective of battery-life in a manual drone is to support the motors and communication devices that allow the user to control the drone remotely. For UAVs, the battery life also must support on-board computation, in addition to these primary objectives. For example, in operations such as search and rescue, UAVs are expected to process the input from the on-board camera to detect and recognize objects or people. Most commonly, the on-board computing framework on a UAV would consist of general-purpose devices, such as multicore CPUs and micro-controllers that consume low power (Vaddi et al., 2021). The high computational requirements of object recognition and vision techniques slowly gave rise to parallel computing architectures. In present day, object recognition and detection are typically done on a Graphical Processing Unit (GPU) using Convolutional Neural Networks (CNNs) (Tuśnio & Wróblewski, 2021). Running computations and algorithms based on CNNs requires significant processing power – for example, the experiments in Tuśnio & Wróblewski (2021) recommend a minimum specification including graphics card NVIDIA GeForce GTX 1060 6 GB (NVIDIA GeForce RTX 2060 6 GB), processor Intel i5 or AMD Ryzen 5 (Intel i7 or AMD Ryzen 7), 8 GB RAM (16 GB RAM).

Thus, the on-board computation needs to be real-time, and it also has to release output in real-time thereby leading to high power consumption. High power consumption could further lead to decaying the system battery, resulting in incomplete missions (Ehsan & McDonald-Maier, 2009). Therefore, the ideal computation algorithm and hardware architecture in UAVs would consume low power while providing real-time solutions, to ensure maximum flight time (Ehsan & McDonald-Maier, 2009). The architecture must work in low-cost frequency as dynamic power

consumption is proportional to clock frequency, which limits UAVs capabilities even more (Ehsan & McDonald-Maier, 2009).

To ensure flight longevity, safety, and autonomous capability, UAVs are typically equipped with on-board cameras and sensors but not much processing capabilities. Therefore, it has been common consensus that the small size of UAVs and limited resources onboard makes them less suitable to process image/video in real time (Ehsan & McDonald-Maier, 2009). Fundamentally, the challenges of deploying compute-intensive, intelligent, vision and object detection capabilities on a UAV platform can be identified as the following from Vaddi et al. (2021) –

a. Keeping power consumption at a minimum to minimize its effect on battery power draw and overall flight time of the drone

b. Minimizing the memory footprint and computational power so as to not over-run the on-board processing capabilities of the drone, and

c. Fast, efficient, real-time data processing and analysis of input data as captured by the on-board camera in critical tasks.

Given the limitations of UAVs for on-board processing of compute-intensive tasks, the idea of *offloading* computation and processing of image/video to a server that is able to run the required computations and send the final results back to the UAV has gained a lot of traction in recent years as many leading companies such as Google, Facebook, Amazon and Huawei have launched their projects to support the application of UAV enabled MEC network (Zhou et al., 2020).

In the next section, we look at the method applied to obtain the desired resources for this research.

Methodology

The objective of this paper is to present with a literature survey of implementations that offload on-board computations from the UAV to edge servers and finally present in-detail one of the many cases available in literature that explain the concepts we talked about in the beginning chapters. In order to accomplish this goal, we first start by obtaining the vast literature available on UAVs, their applications and offloading computation from the UAVs to the edge or cloud servers.

In order to obtain resources specifically related to computation offloading from UAVs, we used Google scholar to search for articles. A combination of the following keywords was used to obtain results for our search: *UAV computation offloading, edge computing and UAVs, cloud computing and UAVs*. The initial result obtained was an overwhelmingly large collection of research, and we had to further filter down the search for reviewing articles. Shown below, in figure 4 is an image of our initial search results, in which we obtained 8,430 results in Google scholar.

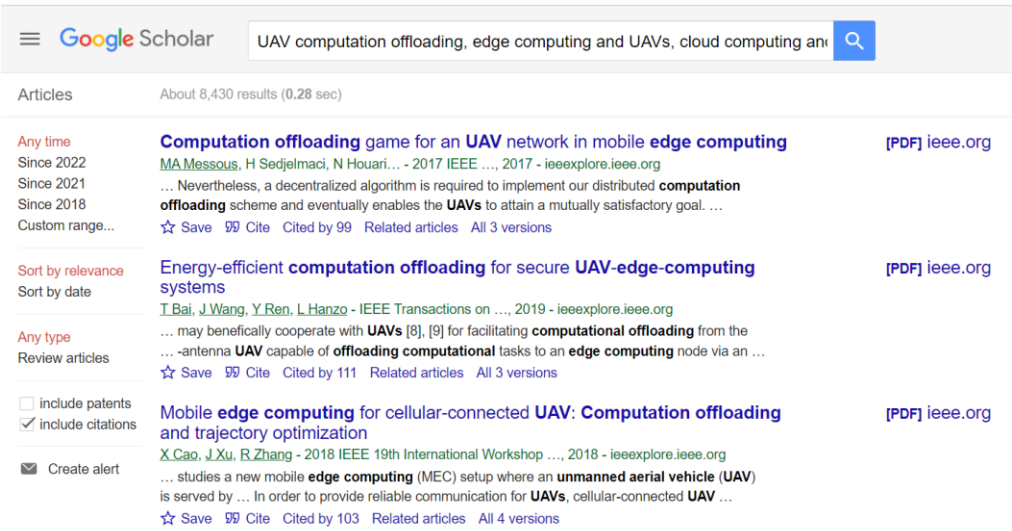

Figure 4: Preliminary Google scholar search results

In the interest of time, we had to narrow down the list of journal articles. While there were articles that were based on offloading computations for vehicles and swarm of drones, our goal was to use articles based only on single UAVs with computation offloading. Moreover, we are particularly interested in articles related to offloading onboard data by leveraging edge computing, but not simply relying on edge servers to maintain communication between the user and the UAV. To ensure we are obtaining the relevant articles, we explored adding on specific keywords such as: *UAV, UAV computation, edge computing* to further narrow down the articles. Additionally, we brought our scope down to include articles in the last 15 years, and as shown in the figure 5 below, we are now able to get the number of articles down to 93 articles.

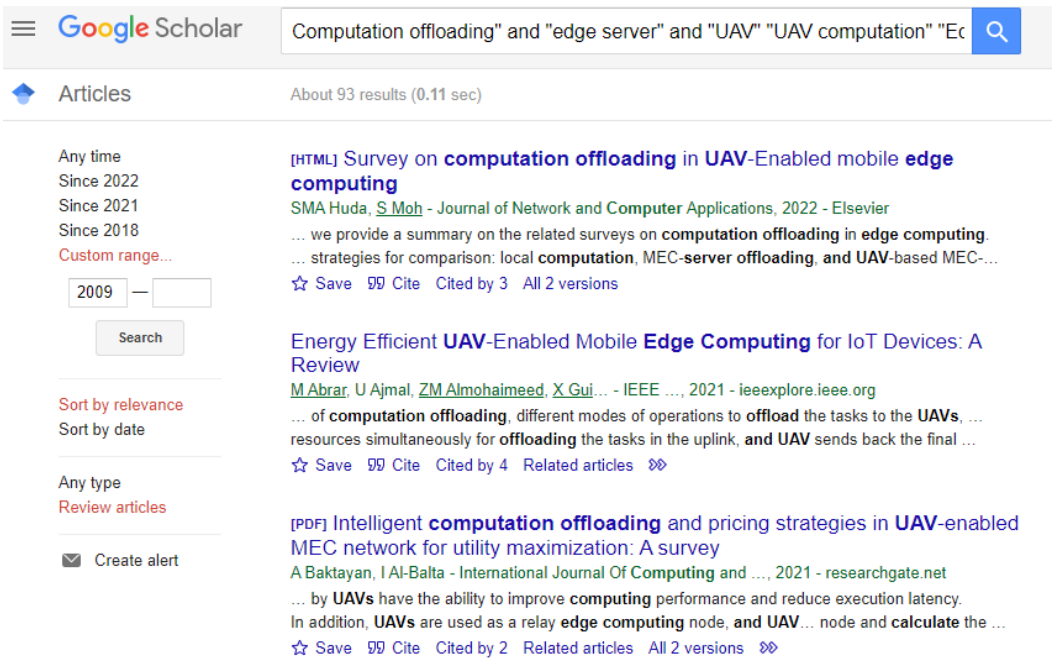

Figure 5: Narrow scope search results

.

At this point, we had to ensure that our sources are reliable and well grounded. So, we only decided to choose our sources from conferences such as IEEE and verified institutions such as

NASA and Cornell University. Then, after narrowing down the articles furthermore, we proceeded to carefully filter through each article, extensively reading their abstract, goals, results and conclusion. This helped us identify research articles most relevant to offloading UAV computation to the cloud or edge servers and their implementations. We also chose a set of three papers to compare and contrast that leveraged the cloud or edge servers to offload UAV computation in various application environments. We list the comparison of these papers in the results and discussion section of this work, but in the next section, we present in detail about computation offloading and mobile edge computing.

Background

Computation offloading

Computation offloading is an approach in which resource-intensive computational tasks to be performed by a system can be "offloaded" to be performed at a "remote" server, typically with more computational capabilities than the system itself (Lin et al., 2019). For example, in the case of the UAVs, the drones themselves may not have the computational ability as would a server or a laptop with faster, more powerful features such as a graphics card, memory and processor capabilities. In such cases, depending on the application itself, many researchers have proposed to take advantage of the resources available in the cloud, or at the edge of the network, i.e., a server or a laptop closer to the end-user, or the device which is offloading computations to the server (Lin et al., 2019). The advantages of computation offloading are that in this case, the UAV itself is not expected to have the on-board computational power to process vision-based or CNN-based algorithms, which are offloaded to edge servers or the cloud. The tradeoff for computational offloading is the additional latency that can be incurred due to the transmission of the input from the camera on-board to the remote server and obtaining the result from the remote server back to the drone. However, since the remote server does have processing capabilities that are much better than on-board processing capabilities, we can also argue that the computations may not take as much time as they take when processed on-board the UAV itself.

Computational tasks can be offloaded to the cloud, or to the edge servers. The main difference between these approaches would be seen in the latency of sending the input from the drone to the cloud or the edge and receiving the result of the processing back from the cloud or the edge server (Zhou et al., 2020). The cloud devices are located farther away from the UAV whereas the edge servers are closer to the UAV. Both cloud and edge servers are computationally stronger

than on-board processing available on the drone, with the cloud being the most capable processing capabilities, memory and network requirements. In figure 6 shown below, we show the interaction between end-users, the edge and the cloud layers for further clarity.

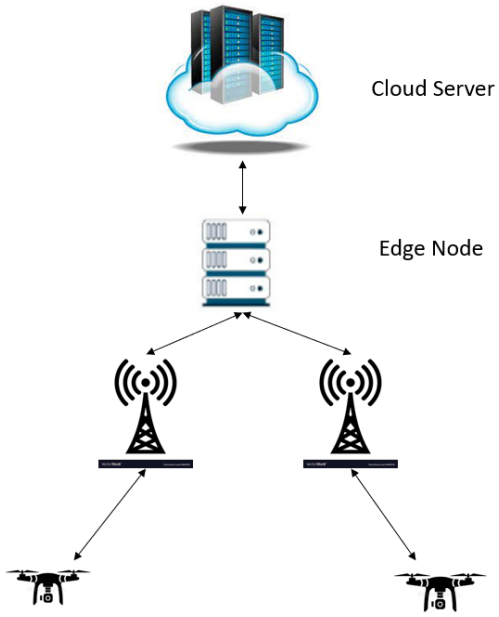

Figure 6: Interaction between the cloud and edge layers

Mobile Edge Computing

According to Zhou et al (2020), "Mobile Edge Computing (MEC) is a promising technique that enables mobile users to offload partial or complete computation intensive tasks to MEC servers for computing". MEC is a way to tackle the challenges of latency in using data centers in cloud and has gained great attention from both industry and academia (Zhou et al., 2020). Recent UAVs have been integrated with an MEC architecture where the UAV is considered as an input/output endpoint that gathers computation task, and relays the computation task to an MEC server for executing computation tasks (Zhou et al., 2020).

There are three basic requirements for the entire system to work: connectivity, on-board sensors, and edge server with an on-board computer. In figure 7, the overview of the MEC system is displayed. Current UAVs mostly use short-range local network techniques such as Wi-Fi, but with the recent rapid development of cellular networks (LTE and 5G), UAVs connected with cellular network have provided a much better solution to connect with edge servers (Chen et al., 2021). The onboard camera and other sensors play a key role in supporting autonomous navigation and obstacle avoidance. From the data captured by the camera and sensors, the system must extract auxiliary information such as the UAV position and 3D map features of the explored areas (Chen et al., 2021). After the auxiliary information is computed in the edge servers, the edge server must map a path for the UAV to travel. For this, the edge server must also have a vision processing algorithm such as simultaneous localization and mapping (SLAM), along with real-time communication link to guarantee faster map update rate (Chen et al., 2021).

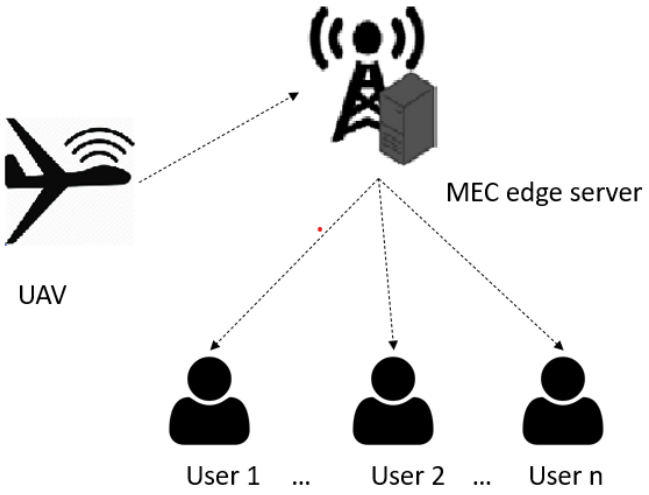

Figure 7: UAV and edge server interaction

If the computation is high, the delay in communication would increase and vice versa. Based on this concept, there are three modes of computation offloading to MEC servers: no offloading, partial offloading, and full offloading (Hayat et al., 2021). With full offloading, the edge server executes the entire vision processing pipeline, which requires the transfer of full images and other data to the server. This results in little to no computation burden onboard and high communication demand (Hayat et al., 2021). With partial offloading, the UAV detects and

tracks image features on board, then transfers the features to the edge servers to finish the remaining computation such as filtering and mapping the path (Hayat et al., 2021). Finally, with no offloading mode, the edge server is only responsible to keep steady communication and provide real time vision and data to the base station, while the UAV does all the computation (Hayat et al., 2021). One of the major benefits of edge computing is low latency, given that the whole process depends upon computing latency and transmission latency. With edge computing, the computing latency is significantly decreased while also providing an ideal tradeoff for transmission latency with the aim to minimize total latency (Yu et al., 2018).

The architecture of edge computing also offers lower bandwidth consumption as the data is processed and compressed in the edge server before sending it to the cloud server or end-users (Yu et al., 2018). It goes without saying that edge computing maximizes the lifetime and energy of UAVs as the computation is offloaded to edge servers. With advantages, there are also challenges and open issues that exist in edge computing. The architecture of edge computing is mainly proposed for networks with either one UAV or one user (Zhou et al., 2020). According to the authors of Zhou et al (2020), since the operation time and battery of UAV are limited and normally a large number of users need to be served, researchers find it challenging to design efficient resource allocation schemes for UAV enabled MEC networks with multiple users and multiple UAVs (Zhou et al., 2020). The other challenge is to jointly optimize the trajectories of multiple UAVs in order to expand the coverage area and improve the computation performance (Zhou et al., 2020). Moreover, there have not been much studies on the security issues in UAV enabled MEC networks (Zhou et al., 2020).

We will, next, discuss about one of the most feasible implementation of computation offloading called DeepBrain.

Implementation Case Study – DeepBrain

While there are many implementations of UAV computation offload to edge computing, we choose to discuss and present the research from DeepBrain as a case study. DeepBrain is a solution to the computation offloading of heavy and intensive computations tasks from a small UAV to the cloud system to reduce energy consumption and therefore, extending the mission lifetime of the UAV (Koubaa et al., 2020). The choice of this particular research article is motivated by the interesting implementation that analyzes and compares the two approaches to performing computations for UAV applications, while also discussing the types of offloading possible.

DeepBrain is mostly used for deep learning applications and leverages cloud resources. The DeepBrain architecture consists of four layers (Koubaa et al., 2020):

- The unmanned system layer,
- The edge layer,
- The cloud layer and
- The end user-layer.

The unmanned system layer is the UAV layer with high resolution camera responsible for aerial image collection on-site and wireless communication interface responsible to communicate with edge or cloud servers and user cellular (4G/5G) networks (Koubaa et al., 2020). The edge layer consists of edge servers used to reduce the load of main cloud servers and are located closest to the UAVs with the purpose of decentralizing the computing among multiple servers (Koubaa et al., 2020). The cloud layer contains the servers that do extensive computing and has storage resources that cannot be handled by the edge servers (Koubaa et al., 2020). The cloud layer also offers drones and user management capabilities to ensure their connectivity, communication,

authentication and availability of services (Koubaa et al., 2020). Finally, the end user layer represents the end-users who are using the DeepBrain system through the internet using web APIs (Koubaa et al., 2020). They interact with UAVs and monitor their states at real time while receiving real time video stream broadcasted from their UAVs after being processed through the DeepBrain architecture (Koubaa et al., 2020).

The major advantage of DeepBrain is that it eliminates the need for the pilot to be in communication range of the UAV during the mission (Koubaa et al., 2020). Users can navigate and monitor their UAV through the internet from any part of the world. It also allows a scalable computation offloading to the edge/cloud to promote deep learning applications even for low-cost UAVs (Koubaa et al., 2020).

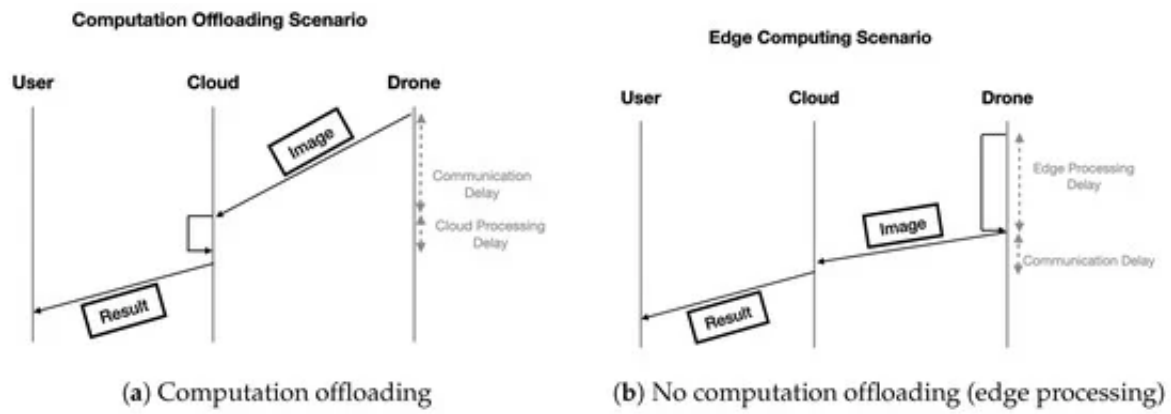

Figure 8: Two offloading approaches from DeepBrain (Koubaa et al., 2020).

DeepBrain uses two computation approaches to provide the user with the advantages mentioned formerly. In the first approach, deep learning computation is completely offloaded to the cloud using video streaming. According to Koubaa et al (2020), "in this approach, the communication cost is higher than the cloud processing cost since messages exchanged between the UAV and the cloud will carry image data, which typically has a much larger size and consumes greater bandwidth and computation cost is smaller because cloud servers use high-performance GPUs."

In the second approach, some deep learning computations are performed at the edge using devices with embedded GPUs. According to Koubaa et al (2020), “in this approach, the cost of computation is higher than the communication cost because edge servers use CPUs or low-cost embedded GPUs to process images with deep learning models and then send the results to the cloud.” Both these approaches resemble the full offloading and partial offloading modes of edge computing that we discussed in the previous section and in figure 8 shown above, we can see the tradeoff between communication delay and processing delay in both approaches.

In DeepBrain, both approaches have their own limitations. It may not be possible for low capability drones to have sufficient network resources making the computation offloading ineffective and so compression might have to be applied before offloading the data (Koubaa et al., 2020). However, the problem with compression is that there might be loss of important features that could be crucial for deep learning models to extract objects and classify them (Koubaa et al., 2020). The other limitation is that full computation offloading with high quality video may incur end-to-end delays up to five seconds and the latency will also be limited because in the case of computing in edge servers on UAV side, the throughput will always be limited to an average of one frame per second (Koubaa et al., 2020).

The next section contains our results and discussion, where we compare DeepBrain with some other implementation case study.

Results and Discussion

Offloading UAV computations is ongoing research whose goal is to minimize latency and cost while maximizing efficiency. To sum up, there are three basic requirements for UAV to offload computation in MEC. They are good cellular network connectivity, an onboard camera or other sensors, and an edge server with enough computing capabilities to perform vision processing algorithms and keep steady communication (Chen et al., 2021). With DeepBrain, there is a debate about whether the computation should be embedded at the edge level, or it should be offloaded to the cloud (Koubaa et al., 2020). The current implementation offloads it to the cloud and in this work, we present the requirements and performance of system alongside the tradeoff between computation and communication latency.

Apart from DeepBrain, there exist other implementations as well that enable offloading UAV computation to the edge. The implementations primarily differ on the basis of their application, offloading approach and the onboard devices on the UAV. In this work, we will briefly compare three shortlisted articles most relevant to the implementation discussed herein on offloading UAV computations to the edge, not including the implementation presented in detail. We also compare these implementations with DeepBrain and the summary of the three implementations are provided below in Table 1.

The first shortlisted article discussed in this paper shows how edge computing can reduce the demand of per-drone bandwidth for video analytics providing timeliness and accurate results (Wang et al, 2018). The second shortlisted article discusses an experimental platform called Hydra – a middleware architecture that enables the adaptive distribution of computation tasks within infrastructure assisted UAV systems (Callegaro et al., 2020). The third shortlisted article provides

a co-operative air-ground solution to target tracking and surveillance using UAV, ground camera, and ground computing system (Liu et al., 2019).

DeepBrain is used for deep learning applications such as search and rescue, vehicle detection, counting and intelligent transportation system (Koubaa et al., 2020). The shortlisted articles are all used for applications such as object detection, target tracking, and real time video analytics. However, there are significant differences between the requirements, implementations and applications that they support. For example, while DeepBrain, typically, does not support small drones, Wang et al (2018)'s application is to perform real time video analytics using small drones. DeepBrain does not require flash storage, HDMI to USB converter, or extensive core processor and GPUs unlike the applications mentioned in Table 1 below. Moreover, DeepBrain supports both full and partial offloading providing the user the flexibility to trade-off between computing and communication latency (Koubaa et al, 2020). But implementations presented in Table 1 supports either full offloading or partial offloading only and limits the system performance.

Table 1: Comparison of similar UAV computation offloading literature

Research	Summary	Offloading Approach	Application	Edge/Onboard Devices
(Wang et al., 2018)	The research provides bandwidth-efficient architecture for small drones that enables live video analytics using mobile edge computing.	Full offloading	Real time video analytics using small drones	High resolution camera and flash storage to preserve captured video
(Callegaro et al., 2020)	The research explores Hydra, a middleware architecture enabling the adaptive distribution of computation	Partial Offloading	Video Analysis and object detection that requires Deep Neural Networks (DNN)	GoPro Hero 4 camera, Magwell HDMI to USB

	tasks within infrastructure assisted UAV systems.			converter, Nividia Jetson Nano card with 4GB RAM, Quad core ARM Cortex-A57 MP Core processor and 128-Core Nvidia Maxwell GPU
(Liu et al., 2019)	The research provides solutions to target tracking using the cooperation with ground surveillance camera and with the assistance of mobile edge computing.	Full offloading	Target tracking and surveillance for social security	High resolution camera

Conclusion

To summarize, UAVs depend on battery life and being lightweight for far-reaching applications. However, in cases where computational tasks are involved, on-board computation has the disadvantages of adding weight to the UAV itself. Thus, more recently, offloading UAV computational tasks to edge servers has been seen as an efficient way of increasing UAV flight longevity. However, offloading to the edge also introduces additional communication latency. In our work presented here, we conclude that the user requirements and system needs will ultimately decide the ideal tradeoff between computing latency and communication latency.

This paper is a preliminary step towards implementation in a research application in the future. Our study provides a detail explanation of computation offloading implemented for UAV computations with applications. Our study is also limited to the overview of implementations such as DeepBrain and does not provide the detail algorithm or any analysis about how it works. Most importantly, since this research is in its early stage, we could not find any studies about the security issues in UAV enabled MEC network. Our work can further be extended by including a study on the security issues in UAVs, analyzing the offloading approaches by implementing the offloading strategies.

References

A drone's components: Guide for beginners. DroneforBeginners. (2021, February 28). Retrieved April 20, 2022, from https://droneforbeginners.com/drones-components/

Anatomy of a drone infographic. Anatomy of a Drone Infographic. (n.d.). Retrieved April 10, 2022, from https://www.dronefly.com/the-anatomy-of-a-drone

Callegaro, D., Baidya, S., & Levorato, M. (2020). Dynamic distributed computing for infrastructure-assisted autonomous UAVs. *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. https://doi.org/10.1109/icc40277.2020.9148986

Chen, Q., Zhu, H., Yang, L., Chen, X., Pollin, S., & Vinogradov, E. (2021). Edge Computing Assisted Autonomous Flight for UAV: Synergies between Vision and Communications. *IEEE Communications Magazine*, *59*(1), 28–33. https://doi.org/10.1109/mcom.001.2000501

Chadha A. (2020). Vision Based Localization of Drones in a GPS Denied Environment. http://hdl.handle.net/10919/99887

Ehsan, S., & McDonald-Maier, K. D. (2009). On-board vision processing for small uavs: Time to Rethink strategy. *2009 NASA/ESA Conference on Adaptive Hardware and Systems*. https://doi.org/10.1109/ahs.2009.6

Hayat, S., Jung, R., Hellwagner, H., Bettstetter, C., Emini, D., & Schnieders, D. (2021). Edge computing in 5G for drone navigation: What to offload? *IEEE Robotics and Automation Letters*, *6*(2), 2571–2578. https://doi.org/10.1109/lra.2021.3062319

Hayden, L. (2021, November 22). *Agriculture drone: How drones are making Agriculture Easier*. The WiredShopper. Retrieved April 11, 2022, from https://thewiredshopper.com/agriculture-drone/

Intelligence, I. (2022, April 15). *Drone market outlook in 2022: Industry growth trends and forecast*. Insider Intelligence. Retrieved April 13, 2022, from https://www.insiderintelligence.com/insights/drone-industry-analysis-market-trends-growth-forecasts/

Koubaa, A., Ammar, A., Alahdab, M., Kanhouch, A., & Azar, A. T. (2020). DeepBrain: Experimental Evaluation of cloud-based computation offloading and Edge Computing in the internet-of-drones for Deep Learning Applications. *Sensors*, *20*(18), 5240. https://doi.org/10.3390/s20185240

Lin, L., Liao, X., Jin, H., & Li, P. (2019). Computation offloading toward Edge Computing. *Proceedings of the IEEE*, *107*(8), 1584–1607. https://doi.org/10.1109/jproc.2019.2922285

Liu, Y., Zhu, C., Deng, X., Guan, P., Wan, Z., Luo, Jie., Liu, E., Zhang, H. (2019). UAV-aided urban target tracking system based on edge computing. *arXiv,* https://doi.org/10.48550/ARXIV.1902.00837

McRae, J. N., Gay, C. J., Nielsen, B. M., & Hunt, A. P. (2019). Using an unmanned aircraft system (drone) to conduct a complex high-altitude search and rescue operation: A case study. *Wilderness & Environmental Medicine*, *30*(3), 287–290. https://doi.org/10.1016/j.wem.2019.03.004

Samad, A. M., Kamarulzaman, N., Hamdani, M. A., Mastor, T. A., & Hashim, K. A. (2013). The potential of Unmanned Aerial Vehicle (UAV) for civilian and mapping application. *2013 IEEE 3rd International Conference on System Engineering and Technology*. https://doi.org/10.1109/icsengt.2013.6650191

Tuśnio, N., & Wróblewski, W. (2021). The efficiency of drones usage for safety and rescue operations in an open area: A case from Poland. *Sustainability*, *14*(1), 327. https://doi.org/10.3390/su14010327

Vaddi, S., Kim, D., Kumar, C., Shad, S., & Jannesari, A. (2021). Efficient object detection model for real-time UAV application. *Computer and Information Science*, *14*(1), 45. https://doi.org/10.5539/cis.v14n1p45

Wang, J., Feng, Z., Chen, Z., George, S., Bala, M., Pillai, P., Yang, S.-W., & Satyanarayanan, M. (2018). Bandwidth-efficient live video analytics for drones via edge computing. *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. https://doi.org/10.1109/sec.2018.00019

Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A survey on the edge computing for the internet of things. *IEEE Access*, *6*, 6900–6919. https://doi.org/10.1109/access.2017.2778504

Zhou, F., Hu, R. Q., Li, Z., & Wang, Y. (2020). Mobile edge computing in Unmanned Aerial Vehicle Networks. *IEEE Wireless Communications*, *27*(1), 140–146. https://doi.org/10.1109/mwc.001.1800594