Mobile-Cloud Cooperative Deep Learning Platform for Mixed Reality Applications

Juheon Yi
juheon.yi@hcs.snu.ac.kr
Seoul National University, Seoul, Korea

ABSTRACT

Mixed Reality (MR), an environment that seamlessly combines the real and virtual worlds to enable real-time interaction between the physical and digital objects, is expected to be a genuinely transformational technology that will change our lives with unprecedented applications. Despite their vast potential, truly immersive MR apps are yet to be developed. The core challenge lies in the unique workload of seamlessly combining virtual information over the physical world with resource-constrained mobile/wearable devices. While such workload often requires a continuous and simultaneous execution of multiple Deep Neural Networks (DNNs) and rendering tasks, existing mobile deep learning platforms are ill-suited as they are mostly designed for running only a single DNN. In this paper, we characterize the workloads of emerging MR apps, comprehensively analyze the requirements and technical challenges, and introduce our research vision and recent projects to develop core mobile deep learning systems to support them.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**.

KEYWORDS

Mixed Reality, Mobile Deep Learning, Cloud Offloading

ACM Reference Format:
Juheon Yi. 2021. Mobile-Cloud Cooperative Deep Learning Platform for Mixed Reality Applications . In *Students in MobiSys (SMS '21), June 24, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3469262.3472387

1 INTRODUCTION

Mixed Reality (MR) apps are getting increasing attention, with the expected market size of $209 billion in 2025 [1]. The unprecedented life-immersive user experiences accelerate the penetration of MR apps into various domains including security, commerce, and education. Also, new forms of MR devices (e.g., Microsoft HoloLens 2, Magic Leap One) are emerging. Despite the huge potential, truly immersive MR apps are yet to be developed.

SMS '21, June 24, 2021, Virtual, WI, USA

ACM ISBN 978-1-4503-8602-9/21/06…$15.00
https://doi.org/10.1145/3469262.3472387

(a) Criminal chasing. **(b) Immersive online shopping.**

Figure 1: Multi-DNN MR application scenarios.

The core challenge of MR apps lies in the unique workload of seamlessly combining virtual and the physical world with resource-constrained MR devices (e.g., mobile/wearables). Specifically, an MR app first needs to accurately analyze the physical world and user behaviors (e.g., gestures, head movements) to decide which virtual contents to generate and where to display them. Such analysis often requires a continuous and simultaneous execution of multiple Deep Neural Networks (DNNs) on high-resolution vision and sensor data streams. Second, the app should seamlessly synthesize and render the virtual contents (e.g., 3D objects, avatar's hand gestures) over the analyzed scenes for immersive user experience. Finally, background DNN computation and foreground UI rendering should be simultaneously performed in real-time over mobile/wearables devices with resource constraints.

Our research vision is developing a mobile-cloud cooperative deep learning platform for fully-immersive MR apps. Specifically, our research goal is to design futuristic and innovative MR apps, comprehensively analyze their workloads, and build core mobile systems to support them. In this paper, we will depict emerging multi-DNN-enabled emerging MR app scenarios, analyze their workload and technical challenges, and introduce our research vision and recent projects to realizing it.

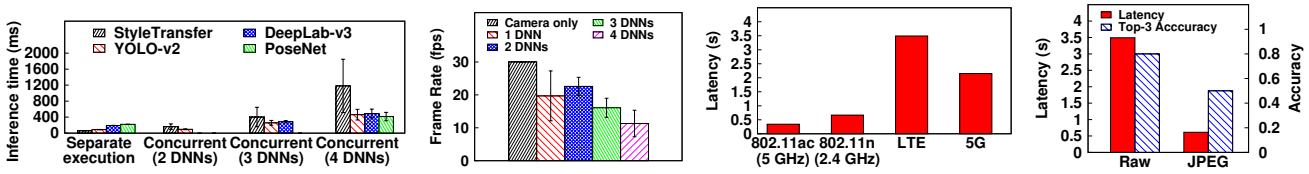
2 APPLICATIONS AND REQUIREMENTS

2.1 Criminal Chasing in Crowded Urban Spaces

Scenario (Figure 1(a)). A police officer chasing a criminal in a crowded space (e.g., shopping mall) sweeps his mobile device to take a video of the area from distance. The mobile device processes the video stream to detect faces and find the matching one with the criminal. Specifically, it continuously runs face detection per each scene image, and face recognition per each detected face. Detection results are seamlessly overlayed on top of the camera frames and rendered on screen to narrow down a specific area to search.

DNN Requirements: face detection, recognition (< 1s per scene).

Rendering Requirements: camera frames (1080p, 30 fps), face bounding boxes and recognition results.



(a) Multi-DNN GPU contention. (b) Rendering-DNN GPU Contention. (c) Offloading latency (raw image). (d) Offloading accuracy in LTE (raw & JPEG compression).

Figure 2: Challenge analysis for future MR workload.

2.2 Immersive Online Shopping

Scenario (Figure 1(b)). An online shopper wearing AR glasses positions a virtual couch in his room to see if the couch matches well before buying it from an online shopping mall. The AR glasses analyze the room by detecting its layout and furniture, and renders the couch in a suitable position. The user can also change the style of the couch (e.g., color, texture), as well as adjust the location with his hand movements. This app requires i) running object detection and image segmentation simultaneously to analyze the room, ii) running hand tracking and image style transfer to recognize user's hand movements and adjust the style of the couch, and iii) rendering the virtual couch on the right spot seamlessly.

DNN Requirements: object detection, image segmentation, hand tracking (1-10 fps), image style transfer (< 0.1s response time).

Rendering Requirements: virtual couch (1,440p, 60 fps).

3 CHALLENGES

The core requirement of the above illustrated emerging MR apps is the concurrent execution of multiple DNN and rendering tasks to accurately analyze the physical world and user behaviors, and combine the virtual contents with the physical world. These requirements are clearly distinguished from prior works [2, 3] that have mostly considered running a single DNN (over a small resolution image (e.g., 300×300)) in an isolated environment without any system resource contention. Such workload incurs the following critical technical challenges on resource-constrained mobile devices (especially the mobile GPU).

3.1 Multi-DNN GPU Contention

Existing mobile deep learning frameworks are mostly designed to run only a single DNN. The only way to support multiple DNNs concurrently is to launch multiple inference engine instances (e.g., TF-Lite's Interpreter, MACE's MaceEngine) on separate threads. However, multiple DNNs competing over limited mobile GPU resources incur severe contention, unexpectedly degrading the overall latency. More importantly, uncoordinated execution of multiple DNNs makes it difficult to guarantee performance for mission-critical tasks with stringent latency constraints.

To evaluate the impact of multi-DNN GPU contention on latency, we run 4 DNNs in the immersive online shopping scenario (Section 2.2) on OpenCL-based Xiaomi MACE framework over LG V50 GPU (Qualcomm Adreno 640). Figure 2(a) shows that with more number of DNNs contending over the mobile GPU, the inference times increase significantly compared to when only a single DNN is running (denoted as Separate execution). More importantly, note that the individual DNN inference times are sufficient to satisfy the app requirements (i.e., the sum of the inference times of 4 the DNNs are 560.02 ms, indicating that they can run at ≈2 fps when coordinated perfectly). However, the uncoordinated execution makes the performance of individual DNNs highly unstable (e.g., the latency of StyleTransfer increases from 59.93±3.68 to 1181±668 ms when 4 DNNs run concurrently), making it highly challenging to satisfy the latency requirement.

3.2 Rendering-DNN GPU Contention

More importantly, existing frameworks only consider a single DNN running in an isolated environment (i.e., no other task contending over the mobile GPU), and are ill-suited for AR apps that require concurrent execution of rendering in presence of multiple DNNs. Figure 2(b) shows the 1080p camera frame rendering rate in presence of the 4 DNNs specified in Figure 2(a). As more number of DNNs are running, rendering frame rate drops significantly due to the similar GPU contention as in Section 3.1, as low as 11.99 fps when all 4 DNNs are running. To make matters worse, GPU contention incurs frame rate heavily fluctuating over time, significantly degrading the perceived rendering quality to users.

3.3 Offloading Latency and Accuracy

One possible option to resolve the aforementioned resource contention is to offload the DNN tasks to the cloud/edge servers. However, offloading the DNN tasks on high-resolution video stream with low latency and high accuracy is highly challenging. For example, Figure 2(c) compares the end-to-end latency to offload the 1080p scene image in raw format from Google Pixel 3 XL to desktop server with RTX 2080 Ti GPU for the multi-DNN face identification pipeline in the criminal chasing scenario (Section 2.1) under different networks: outdoor LTE (11 Mbps) and 5G (45 Mbps), indoor 802.11n (92 Mbps) and 802.11ac (292 Mbps). The offloading latency remains above 0.3 s even for 802.11ac network, and increases to as high 3.4s in outdoor LTE, mainly due to the large data size (i.e., 6 MB 1080p image) to be transferred over the network. While one may think that utilizing image compression (e.g., JPEG) can reduce the network transmission latency, it comes at the cost of DNN accuracy drop as shown in Figure 2(d), as such compression algorithms are mainly designed to minimize the impact on human cognition [4].

4 OUR VISION AND RECENT PROJECTS

Our research vision is developing a mobile-cloud cooperative deep learning platform for fully-immersive MR apps. Specifically, we aim at developing a platform with the following features. First, it intelligently utilizes the mobile and cloud to cooperatively execute

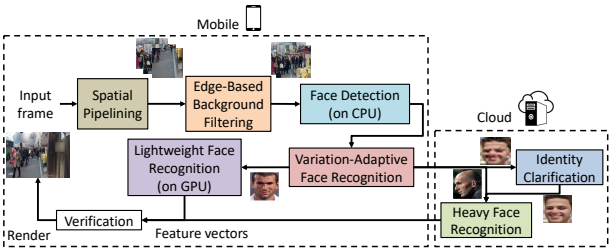


Figure 3: System Architecture of EagleEye.

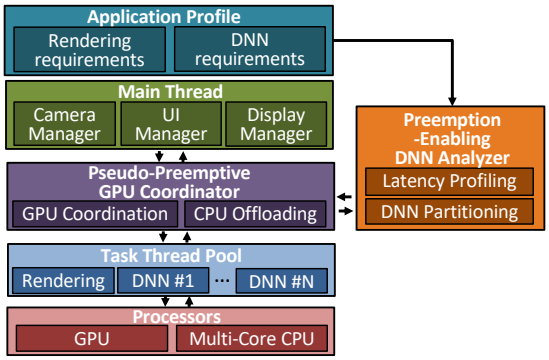


Figure 4: System Architecture of Heimdall.

the MR workload. Second, it is designed with MR-specialized mobile deep learning techniques, Finally, it is composed of comprehensive optimizations across the application, middleware, and OS layer.

In this light, we have worked on a number of challenging research projects to characterize the workload of future MR apps, develop core optimization techniques to support the workload, and design innovative apps on top of them [5–7]. Specifically, we introduce our two recent projects published on ACM MobiCom 2020.

4.1 EagleEye: MR Person Identification in Crowded Urban Spaces

We developed EagleEye [5], an innovative MR system on emerging types of wearable devices capable of finding missing person(s) in crowded scenes. It continuously captures the video of the place using commodity mobile cameras and identifies the target(s) in real-time using a DNN-based face identification technique. The importance of EagleEye lies not only in that it overcomes the fundamental limitations of prior mobile face identification systems capable of identifying only a small number of people in close vicinity, but also in that it characterizes the workload of future MR apps that require repetitive execution of multiple DNNs over high-resolution complex scene images.

Designing EagleEye involves two challenges: (i) accurately recognizing low-resolution faces captured from a distance, and (ii) running the multi-DNN face identification pipeline over high-resolution images in real-time. To tackle the challenges, we first develop an Identity Clarification Network capable of reconstructing high-resolution faces to enhance recognition accuracy. Furthermore, we develop a Content-Adaptive Parallel Execution technique, which adaptively selects different DNN pipelines depending on the scene content and parallelizes the execution over mobile and cloud. EagleEye achieves 9.07× faster latency compared to naive execution, with only 108 KBytes of data offloaded.

4.2 Heimdall: Mobile GPU Coordination Platform for MR Apps

We also innovated existing mobile deep learning frameworks to support emerging MR apps. We have thoroughly analyzed the computational workload of MR apps and developed Heimdall [6], the first mobile GPU coordination platform to support concurrent multi-DNN and rendering tasks on mobile devices. We believe Heimdall will be an important cornerstone that provides useful insights to researchers in designing and implementing many futuristic AR apps. Specifically, the fundamental limitations of existing frameworks (e.g., TensorFlow-Lite) are that (i) they are designed for independent execution of a single, limited type of DNN, and (ii) they cannot support real-time, concurrent multi-DNN execution along with flexible coordination with foreground rendering. To tackle the challenges, Heimdall is empowered by a novel *Pseudo-Preemption* mechanism that breaks down the bulky DNNs into smaller units, prioritizes and flexibly schedules concurrent GPU tasks to satisfy the app requirements. With a simple, yet effective application-level solution, Heimdall enhances the frame rate from ≈12 to ≈30 fps while reducing the worst-case DNN inference latency by up to ≈15 times compared to the baseline multi-threading approach.

5 CONCLUSION AND FUTURE WORK

Despite their vast potential, truly immersive MR apps are yet to be developed. In this paper, we depicted emerging MR app scenarios and their workload, analyzed the technical challenges in realizing them, and introduced our research vision and recent efforts on developing a mobile-cloud cooperative deep learning platform.

To realize our vision, we will continue to design creative and innovative MR apps, comprehensively analyze their requirements, and build core systems to enable them. To better understand and cover more diverse aspects of MR, we will conduct interdisciplinary research at the intersection of mobile computing, machine learning, networking, and HCI. For the next close steps, we are working on a number of projects to extend our systems, which include (i) full-stack MR service design leveraging EagleEye as the underlying platform, (ii) Heimdall extension for emerging neural processor support (e.g., NPU, TPU), (iii) power profiling and optimization for mobile deep learning systems, and (iv) support for diverse sensor inputs and workloads (e.g., spatial audio and 3D vision).

REFERENCES

[1] "Extended Reality (XR) Is The Hot Topic Of 2020 And Beyond: Here's Why," https://www.forbes.com/sites/theyec/2019/07/08/extended-reality-xr-is-the-hot-topic-of-2020-and-beyond-heres-why/?sh=27b67c63a464/. Accessed: 28 May 2021.
[2] L. N. Huynh, Y. Lee, and R. K. Balan, "DeepMon: Mobile GPU-based deep learning framework for continuous vision applications," in *Proc. ACM MobiSys*, 2017.
[3] M. Xu, M. Zhu, Y. Liu, F. X. Lin, and X. Liu, "DeepCache: Principled cache for mobile deep vision," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 129–144.
[4] X. Xie and K.-H. Kim, "Source compression with bounded dnn perception loss for iot edge computer vision," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.
[5] J. Yi, S. Choi, and Y. Lee, "EagleEye: Wearable camera-based person identification in crowded urban spaces," in *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*. ACM, 2020.
[6] J. Yi and Y. Lee, "Heimdall: mobile GPU coordination platform for augmented reality applications," in *Proc. ACM MobiCom*, 2020.
[7] J. Yi, S. Kim, J. Kim, and S. Choi, "Supremo: Cloud-assisted low-latency super-resolution in mobile devices," *IEEE Transactions on Mobile Computing*, 2020.