



Optimizing User Experience in Wearable Cognitive Assistance through Model Specialization

Chanh Nguyen

School of Computer Science, Carnegie Mellon
University
Pittsburgh, PA, USA

Mahadev Satyanarayanan

School of Computer Science, Carnegie Mellon
University
Pittsburgh, PA, USA

ABSTRACT

Wearable Cognitive Assistance (WCA) is a rapidly evolving application that relies on accurate computer vision models for optimal performance and user experience. However, adapting these models to varying user workstation backgrounds can be challenging, as it often necessitates extensive data collection and model retraining. To address this challenge, we propose an approach that focuses on improving model specialization to enhance the accuracy of model inference. Our method eliminates the need to gather the entire training dataset from each individual end user. This not only reduces labor-intensive work but also minimizes bandwidth requirements for transferring data to remote servers for training.

We successfully train specialized models that are tailored to the unique characteristics of each workstation. These specialized models consistently achieve competitive accuracy levels during model inference, comparable to the ground truth models trained with real data collected directly from the workstations, which ultimately enhances the overall user experience with the WCA application.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded and cyber-physical systems**.

KEYWORDS

Wearable Cognitive Assistance, Edge Computing, Edge-native Application, Machine Learning, Model Specialization, Computer Vision.

ACM Reference Format:

Chanh Nguyen and Mahadev Satyanarayanan. 2023. Optimizing User Experience in Wearable Cognitive Assistance through Model Specialization. In *The 2nd Workshop on Smart Wearable Systems and Applications (SmartWear '23)*, October 6, 2023, Madrid, Spain. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3615592.3616850>

1 INTRODUCTION

Wearable Cognitive Assistance (WCA) [5] is an emerging category of applications that integrates wearable devices, computer vision, and edge computing to augment human cognition. WCA applications leverage the capabilities of wearable devices to deliver real-time support and guidance to users, assisting them in performing various tasks or activities. While the concept has been present for almost two decades [16, 17], recent technological advancements in computer vision, edge computing offloading, and the widespread availability of wearable devices have greatly enhanced the accessibility of these applications.

The CMU Living Edge Lab has developed and implemented the *Gabriel platform*¹, an open-source software platform aimed at simplifying the development of WCA applications [5]. Gabriel effectively abstracts away the complexities associated with system-level functionalities commonly required across multiple applications, including network communication and data pre-processing [3]. Figure 1 illustrates the workflow of a WCA application built using the Gabriel platform.

In this workflow, sensor data, specifically video frames, captured by an end-user's wearable device, undergoes initial encoding and compression before being wirelessly transmitted to a *cloudlet*—a server situated in close network proximity. On the cloudlet, the sensor streams are processed by a set of cognitive modules that leverage computationally intensive computer vision models, such as object detection and classification. The cognitive module outputs are then integrated by a task-specific user guidance module, responsible for conducting higher-level cognitive processing. By establishing a mapping between the user's current progress and



This work is licensed under a Creative Commons Attribution International 4.0 License.

SmartWear '23, October 6, 2023, Madrid, Spain

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0343-0/23/10.

<https://doi.org/10.1145/3615592.3616850>

¹<https://github.com/cmusatyalab/gabriel>

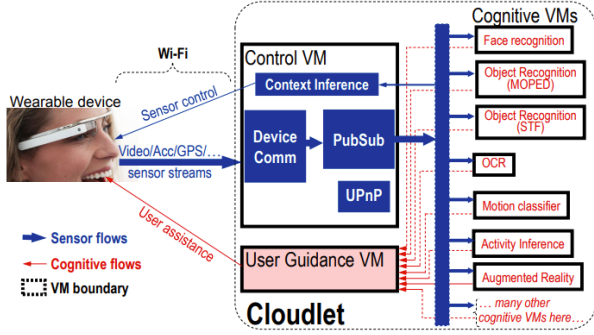


Figure 1: The WCA application architecture developed with the Gabriel platform. Reference source [5].

the outputs obtained from the cognitive modules, the guidance generator triggers task-appropriate visual, verbal, or tactile guidance, which is subsequently transmitted back to the end-user’s wearable device. Thanks to the Gabriel platform, we have extensively delved into the realm of wearable cognitive assistance, resulting in the creation of nearly 20 applications in this domain ².

As presented, the deep neural network (DNN) lies at the core of a WCA application. Therefore, the accuracy of these machine learning models plays a crucial role in determining the success of WCA. However, training these DNN models requires a large number of ground-truth images. Each image in the dataset must be meticulously annotated with bounding boxes or polygons to identify the object of interest. This annotation process is highly time-consuming and labor-intensive, often serving as a significant bottleneck in the development of WCA applications. To address this challenge, we have developed an automatic annotation tool called tinyHulk [10]. The tool automates the annotation process and efficiently generates a clean and high-quality training set specifically from input videos recorded with a green background.

Furthermore, to address resource limitations at the cloudlet and fulfill critical response time requirements, WCA applications strive to deploy lightweight models with lower computational complexity. However, this tradeoff for efficiency can result in a decrease in model inference accuracy, leading to diminished performance. This impact is particularly pronounced in computer vision models, where the models learn to recognize and extract relevant features from images to make predictions. Challenges such as domain shift [7, 15], feature extraction [11], and generalization [1] arise when the backgrounds in the training set significantly differ from those in the input images, further contributing to the performance decline.

In the context of WCA applications, where the user experience is highly dependent on the performance of computer

vision models at specific user workstations, the need for generality is considered unnecessary. In light of this, our paper presents a training method that focuses on enhancing model specialization. By tailoring the model to the specific characteristics of the user workstations, we aim to improve the performance of model inference, ultimately leading to enhanced user experience. We employ the augmentation technique offered by tinyHulk to create a new training dataset where the background of each image is replaced with the background of the user’s workstation. By incorporating this customized training data, the trained model demonstrates a noteworthy improvement in model inference performance specifically at the user’s workstation.

The **key contributions** of this paper are:

- We introduce an approach to generate specialized augmented data for training computer vision models, specifically designed to enhance model specialization within the context of WCA.(Section 3).
- We conduct a comprehensive evaluation to assess the accuracy of the trained model and compare it against a ground truth model trained with real dataset (Section 5).

Our approach facilitates the rapid generation of training datasets, minimizing the time and effort required from humans. Furthermore, the models trained with the augmented data exhibit competitive performance compared to models trained using real data collected from the user’s workstation. As a result, the enhanced models contribute to an improved user experience with WCA applications.

2 RELATED WORK

In this section, we provide a literature review on related work that focuses on improving the inference of DNN models through the enhancement of model specialization.

Model specialization [9, 14] is utilized as a means to optimize inference costs in scenarios where generality is known to be unnecessary. In the field of video analytics for static cameras, Rivas et al. [14] note that a static camera, which maintains a consistent position, orientation, lens, distance, and point of view, continuously captures the same scene. Based on this observation, they propose a Contextually Optimized Video Analytics framework (COVA) to enhance the accuracy of lightweight models. COVA achieves this by specializing and tailoring the models to the specific context in which the cameras will be deployed. Similarly, Shen et al. [18] identify the presence of short-term skews in the class distribution commonly found in everyday videos. They leverage this insight to train models online that are specialized for such distribution patterns. Ravi et al. [9] proposed model distillation to specialize accurate, low-cost semantic segmentation models for a target video stream, enabling

²<https://www.cmu.edu/scs/edgecomputing/resources/videos.html>

more efficient inference by adapting compact models to the specific frame distribution observed by a single camera. Likewise, Khani et al. [6] proposed an Adaptive Model Streaming (AMS) approach to enhance the performance of efficient lightweight models for video inference on edge devices. The AMS approach focuses on specializing a lightweight model for a specific video and task, thus improving the overall performance of the model in edge computing scenarios.

In line with this perspective, in the realm of WCA, users predominantly work on their individual workstations, where the quality of their experience is directly influenced by the accuracy of computer vision model inference in that specific environment. To tackle this, our proposed method focuses on specialized training of the model, tailored specifically to the user’s workstation. We leverage the background replacement augmentation feature of our developed annotation tool, tinyHulk, to generate a new training set: each frame image in this set is modified by replacing the original background with the actual background of the user’s workstation. By incorporating this customized training data, the trained model demonstrates a noteworthy improvement in model inference performance specifically at the user’s workstation.

3 ENHANCING MODEL SPECIALIZATION IN WCA

In this section, we begin by introducing the automatic annotation tool, tinyHulk. Subsequently, we detail the process of utilizing the background replacement feature of tinyHulk to generate a new training set specifically tailored for training the model to specialize in the user’s workstation environment.

3.1 tinyHulk - Lightweight Automatic Annotation Tool

We developed tinyHulk³ [10], an automatic annotation tool that reduces human time and effort for the labor-intensive annotation tasks. tinyHulk is developed using the Open Computer Vision libraries (OpenCV) [12] for computer vision algorithms. The workflow of tinyHulk, as depicted in Figure 2, encompasses the following steps: 1) video parsing and duplicate removal; 2) bounding-box drawing; 3) background replacement; 4) annotation result inspection and modification window; and 5) labeled data generation.

To generate training datasets for the computer vision models, we employ the use of tinyHulk in the following manner. Initially, we record videos that focus on the object of interest, with a green background surface serving as the backdrop. Subsequently, these recorded videos are automatically processed by tinyHulk, resulting in the generation of annotated

image training sets. The output generated by tinyHulk is a high-quality training set comprising clean frames, accompanied by corresponding metadata that accurately describes the bounding box for the object within each frame.

3.2 Enhancing Model Specialization in WCA with tinyHulk

tinyHulk incorporates an augmentation component that utilizes the widely-used chroma-keying technique [2] to generate augmented images. This component replaces the background of an original image with a user-specified background, resulting in a new augmented image.

To generate a training dataset specialized for the user’s workstation, the process involves capturing an image of the background scene where the WCA application will be used. This image is then sent to a cloudlet where tinyHulk is deployed. Within the cloudlet, the original training set, comprising green background frames, along with their corresponding metadata describing the bounding box of the object of interest in each frame, is preserved. Within the augmented dataset, each new frame inherits the bounding box information from the corresponding original frame. Finally, the augmented dataset is combined with the original training set to train a new computer vision model that is specialized to the user’s workstation. This specialized model is subsequently incorporated into the computer vision process of WCA. The process is illustrated in Figure 3.

4 EXPERIMENT

In this section, we describe the experimental setup for developing a Wearable Cognitive Assistance (WCA) application. Specifically, we focus on the computer vision process within the application. We describe how we employed tinyHulk to generate augmented data, incorporating the new workstation background, to train our computer vision models.

4.1 A WCA Application for Assembly Task

We develop a WCA assembly application designed to assist users in completing a specific task of assembling a Meccano set. The sequential assembly process is illustrated in Figure 4.

The application determines the current step of the assembly task displayed in a camera feed and provide the end-user with step-by-step guidance for completing the task. To achieve this, we employ a two-stage vision process inspired by Gebru et al. [4]. For each image frame captured from the end-user’s camera, the initial stage involves identifying the region in the image that contains the assembly being worked on. This is accomplished by utilizing a Faster R-CNN (Region-Based Convolutional Neural Network) model [13], which is specifically built for object detection. Once the region is identified, the image is cropped accordingly.

³<https://github.com/cnguyen123/tinyHulk>

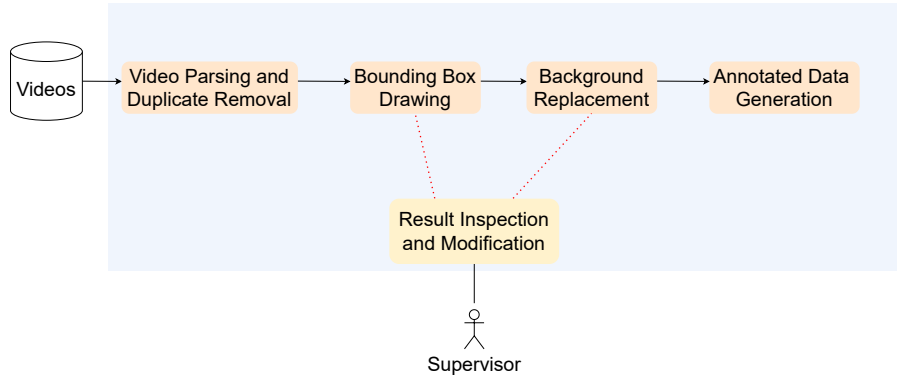


Figure 2: The workflow of tinyHulk.

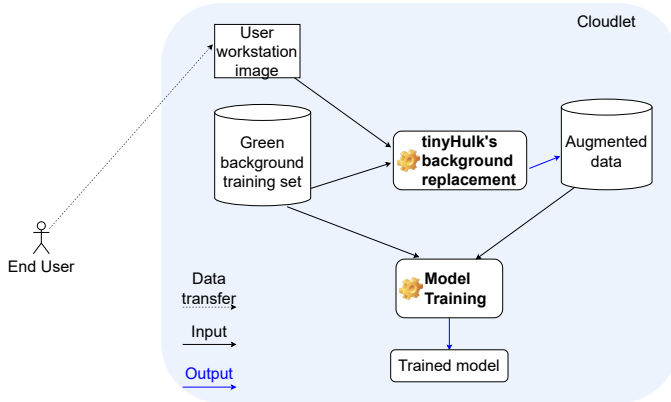


Figure 3: The process of generating augmented data and training specialized models.

In the subsequent stage, the cropped image undergoes classification using the Fast MPN-COV (Matrix Power Normalized Covariance pooling) ConvNet [8]. The Fast MPN-COV model is designed to provide one output label for each step of the assembly task, resulting in six possible outputs for the WCA application in our experiments. The classification result obtained from this model indicates the specific task step depicted in the image frame.

We utilize tinyHulk to generate training datasets for these models, following the description in Section 3.1. Consequently, frames that display the outcome of every step in the assembly task, along with their respective bounding box annotations, are returned and stored on a cloudlet.

Prior to the end-user using the WCA with their own workstation, we request an image of the workstation’s scene where the application will be used. On the server side, tinyHulk assists in generating augmented data by replacing the green background with the provided new background, as depicted in Figure 5. This new data will be incorporated into

the training set to facilitate the retraining of the computer vision process of the application.

4.2 Ground-truth models trained with real data

In addition to the previously trained computer vision process, we further train the alternative models using real data that captures the placement of objects within the new workstation scene. To accomplish this, we involve two individuals, both of whom are familiar with the step-by-step assembly task⁴. To annotate the real data, we employ the use of CVAT⁵, a well-known and widely used annotation tool specifically designed for image data. We employ these models as ground truth references for evaluating and comparing the performance of the computer vision model trained with augmented data.

Overall, Table 1 summarizes the training datasets and their sizes. We implemented 5 distinct computer vision processes, each includes trained models using unique training sets above:

- **Green:** Models trained with the green background training set. It is worth noting that, the original green background encompassed a total of 2158 frames. The model trained on this dataset exhibited inferior performance compared to the model trained using the clean green dataset (572 frames in total), which involved the removal of duplicate and blurred frames. Therefore,



⁴Given the impracticality of expecting users to have prior knowledge on how to perform the step-by-step assembly task and capture the data from the beginning, we acknowledge that these scenarios may not reflect real-world conditions. However, in order to construct a ground truth model that serves as a baseline for comparison with our augmented data-trained model, we assume that the user involved has prior knowledge of the step-by-step assembly task

⁵<https://github.com/openai/cvat>

Table 1: Training sets and their total frames.

Training set	Total Frames
Green	572
Green + Augmented	1144
Real-black	2685
Real-wooden	3077

Table 2: Test sets size and sample.

Test set	Size	Sample
Black table	4871	
Wooden table	5106	

for the purpose of our presentation, we present the trained model using the refined clean green dataset.

- **Specialized_wooden:** Models trained with the green background and the augmented data specifically generated for wooden table workstation.
- **Specialized_black:** Models trained with the green background and the augmented data specifically generated for black table workstation.
- **Real_wooden:** Models trained with the real data collected at the wooden table workstation.
- **Real_black:** Models trained with the real data collected at the black table workstation.

The models were trained on a cloudlet equipped with an Intel Xeon Processor E5-2699 CPU and an Nvidia GeForce GTX 1080 Ti GPU. Specifically, the object detection model Faster R-CNN was trained with 25,000 steps, while the classifier Fast MPN-COV was trained for 50 epochs.

To evaluate the performance of the WCA application using various computer vision processes, we collected test sets where the assembly task was performed on two distinct workstations: the black table, and the wooden table. Each frame in the test set was meticulously labeled with the corresponding step name (e.g., short bar, red bar, screw, long bar, short bar + red bar, and full). Figure 2 presents the test sets size. The accuracy of the WCA’s computer vision process is calculated using the equation provided below:

$$A = \frac{c}{T} * 100 \quad (1)$$

Here, c denotes the total number of frames correctly labeled by the computer vision process, and T represents the total number of frames in the test set.

5 RESULT AND DISCUSSION

This section presents the outcomes of the WCA application when employing different computer vision processes on the designated test set. We emphasize the benefits of model specialization in enhancing the accuracy of computer vision models, ultimately resulting in an enhanced user experience for the WCA application.

5.1 How does the performance of the specialized models compare to the ground truth models?

For each user’s workstation, we train specialized models using both the new augmented data generated by tinyHulk and the existing green data. Additionally, we train real models using manually collected and annotated data from each user’s workstation. Subsequently, we evaluate the performance of these trained models by running the WCA with different computer vision models over the collected test sets. Table 3 provides the accuracy of each computer vision process on a specific test set. We observe that in both test cases, the specialized models outperform the green model, which aligns with our expectations. This can be attributed to two factors. Firstly, the training set of the specialized models is twice the size of the green model’s training set, providing more data for learning. Secondly, the specialized models are trained using augmented data that closely resembles the background of the workstation used to capture the test set. This close resemblance in background enhances the accuracy of the model’s inference.

In both test cases, we have consistently observed that models trained with a training set that incorporates a background similar to that of the test set exhibit higher inference accuracy than those other models. For instance, the specialized_black and real_black models outperformed the specialized_wood and real_wooden models when tested on the black table, and vice versa. These results validate the importance of model specialization in enhancing the performance and accuracy of the DNNs models.

Interestingly, we observe that the specialized model performs exceptionally well, even in comparison to the ground truth model trained with real data. In the case of the test set with a wooden table background, we find that the specialized_wooden model performs almost on par with the ground truth model real_wooden, achieving an accuracy of 83.1% compared to 84.7% of the real_wooden. Particularly, in the black table test set, the specialized_black model surpasses the ground truth real_black, achieving an accuracy of 87.5% compared to 80.9%. Upon reviewing the training set used for the ground truth model in this test set, we identified the presence of duplicate and blurred frames. These factors

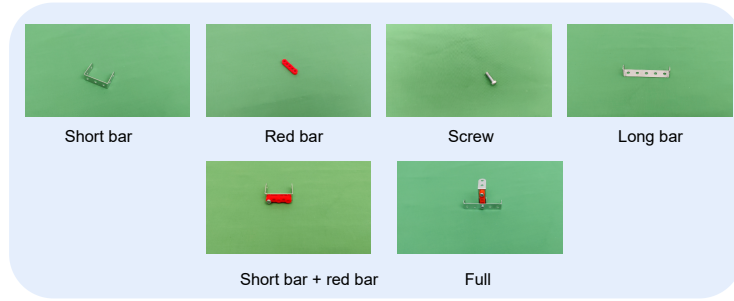


Figure 4: Step-by-step assembly process followed by the WCA in our experiment.

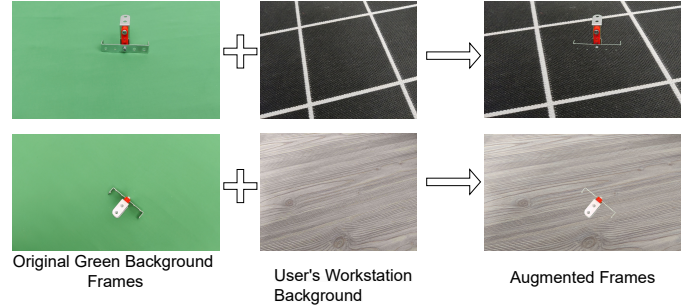


Figure 5: Augmented data generated by tinyHulk (green background replaced with workstation-specific backgrounds).

Table 3: Accuracy of trained computer vision models in WCA across different test sets. The number in bold represents the best result obtained for the test set.

Model	Accuracy(in percentage)	
	Black Table	Wooden Table
Green	15.2	16.3
Specialized_black	87.5	71.4
Specialized_wooden	59.4	83.1
Real_black	80.9	60.5
Real_wooden	64.6	84.7

have contributed to a lower quality training set, resulting in reduced accuracy in the trained ground truth model.

In summary, leveraging model specialization with the assistance of tinyHulk has resulted in a significant improvement in the accuracy of the trained model. The specialized model exhibits a competitive level of accuracy when compared to the model trained with real data.

5.2 How does tinyHulk contribute to time and effort savings in the process of model specialization?

In our experiment, the WCA application is built to assist users in completing the task by providing step-by-step guidance. Since users may not have prior knowledge of the task,

it is impractical to expect them to record a training set at their workstations for training a specialized model for the application. Here, tinyHulk provides a solution that simplifies the data collection process. Users only need to capture a photo of the background of their workstation, and tinyHulk generates augmented data using the original green training set, which has been prepared by professional data engineers in advance. According to the report [10], the process of generating augmented data for training purposes, specifically for the object of interest or each step in WCA, takes approximately 20 seconds, regardless of the size of the training set (the time is dedicated to identifying the HSV threshold required to modify the background of a sample frame. Once determined, this threshold is uniformly applied to the entire dataset). Furthermore, compared to the size of the entire training set consisting of thousands of frames, the individual background photo size (e.g., 3.8MB in the experiment) is significantly smaller, which resulting in substantial bandwidth savings for data transferring from the end-user’s device to the cloudlet.

Our method of model specialization is not limited to a specific WCA application and can be extended to various other use cases that rely on computer vision models. For instance, it can be effectively applied to applications involving object recognition and detection. These applications can benefit from improved model specialization without the need for

labor-intensive human involvement in recording and creating training sets. This not only reduces the workload on humans but also eliminates the risk of generating low-quality training sets due to limited data collection skills or expertise. This is particularly crucial considering our observations in Section 5.1, where low-quality training sets led to lower model accuracy.

6 CONCLUSION

In the context of the Wearable Cognitive Assistance (WCA) application, the accuracy of DNNs’ computer vision models can suffer when deployed on workstations with backgrounds different from those in the model’s training set. To address this challenge, we propose an approach that focuses on improving model specialization to enhance the accuracy of model inference.

Our methodology involves leveraging the augmentation capabilities of our developed annotation tool, tinyHulk, to generate augmented data that is specific to the background of each user’s workstation. This augmented data is then utilized as the training set for the computer vision model in WCA applications. The approach minimizes the need for extensive human effort and time for collecting training data, as well as the bandwidth necessary for transferring data.

The experimental results demonstrate the efficacy of the proposed approach. We successfully train specialized models that are tailored to the unique characteristics of each workstation. These specialized models consistently achieve competitive accuracy levels during model inference, comparable to the ground truth models trained with real data collected directly from the workstations, effectively mitigating the impact of background variations and ultimately enhancing the overall user experience with the WCA application.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and the United States National Science Foundation (NSF) under award number CNS-2106862. The work was done in the CMU Living Edge Lab, which is supported by Intel, ARM, Vodafone, Deutsche Telekom, CableLabs, Crown Castle, InterDigital, Seagate, Microsoft, the VMware University Research Fund, and the Conklin Kistler family fund. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view(s) of their employers or funding sources.

REFERENCES

- [1] BEERY, S., LIU, Y., MORRIS, D., PIAVIS, J., KAPOOR, A., JOSHI, N., MEISTER, M., AND PERONA, P. Synthetic examples improve generalization for rare classes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2020), pp. 863–873.
- [2] BEN-EZRA, M. Segmentation with invisible keying signal. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)* (2000), vol. 1, IEEE, pp. 32–37.
- [3] CHEN, Z. *An application platform for wearable cognitive assistance*. PhD thesis, Ph.D. Dissertation. Carnegie Mellon University, 2018.
- [4] GEBRU, T., KRAUSE, J., WANG, Y., CHEN, D., DENG, J., AND FEI-FEI, L. Fine-grained car detection for visual census estimation. *Proceedings of the AAAI Conference on Artificial Intelligence* (Feb. 2017).
- [5] HA, K., CHEN, Z., HU, W., RICHTER, W., PILLAI, P., AND SATYANARAYANAN, M. Towards wearable cognitive assistance. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services* (2014), pp. 68–81.
- [6] KHANI, M., HAMADANIAN, P., NASR-ESFAHANY, A., AND ALIZADEH, M. Real-time video inference on edge devices via adaptive model streaming. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 4572–4582.
- [7] KONDRATEVA, E., POMINOVA, M., POPOVA, E., SHARAEV, M., BERNSTEIN, A., AND BURNAEV, E. Domain shift in computer vision models for mri data analysis: an overview. In *Thirteenth International Conference on Machine Vision* (2021), vol. 11605, SPIE, pp. 126–133.
- [8] LI, P., XIE, J., WANG, Q., AND GAO, Z. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 947–955.
- [9] MULLAPUDI, R. T., CHEN, S., ZHANG, K., RAMANAN, D., AND FATAHALIAN, K. Online model distillation for efficient video inference. In *Proceedings of the IEEE/CVF International conference on computer vision* (2019), pp. 3573–3582.
- [10] NGUYEN, C., IYENGAR, R., DONG, Q., BLAKLEY, J., AND SATYANARAYANAN, M. tinyHulk: Lightweight annotation for wearable cognitive assistance. Preprint: https://github.com/cnguyen123/tinyHulk/blob/main/Tech_report_compressed.pdf, Feb. 2023.
- [11] NIXON, M., AND AGUADO, A. *Feature extraction and image processing for computer vision*. Academic press, 2019.
- [12] OPENCV. <https://opencv.org/>, Accessed: 2022-09-30.
- [13] REN, S., HE, K., GIRSHICK, R., AND SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).
- [14] RIVAS, D., GUIM, F., POLO, J., SILVA, P. M., BERRAL, J. L., AND CARRERA, D. Towards automatic model specialization for edge video analytics. *Future Generation Computer Systems* 134 (2022), 399–413.
- [15] SAENKO, K., KULIS, B., FRITZ, M., AND DARRELL, T. Adapting visual category models to new domains. In *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV 11* (2010), Springer, pp. 213–226.
- [16] SATYANARAYANAN, M. From the editor in chief: Augmenting cognition. *IEEE Pervasive Computing* 3, 2 (2004), 4–5.
- [17] SATYANARAYANAN, M., BAHL, P., CACERES, R., AND DAVIES, N. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing* 8, 4 (2009), 14–23.
- [18] SHEN, H., HAN, S., PHILIPPOSE, M., AND KRISHNAMURTHY, A. Fast video classification via adaptive cascading of deep models. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 3646–3654.