2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

MAVI: Mobility Assistant for Visually Impaired with Optional Use of Local and Cloud Resources

Rajesh Kedia*[1], Anupam Sobti*[1], Mukund Rungta*, Sarvesh Chandoliya*, Akhil Soni*, Anil Kumar Meena*,
Chrystle Myrna Lobo*, Richa Verma†, M. Balakrishnan*, and Chetan Arora*
*Indian Institute of Technology Delhi †Indraprastha Institute of Information Technology Delhi

***Abstract*—Independent mobility of visually impaired people is key to making an inclusive society for them. Unstructured infrastructure in developing countries pose significant challenges in developing aids to address the mobility problem of visually impaired. Most of the assistive devices available internationally assume a structured and controlled environment severely restricting the applicability of such devices. In this paper, we assess the ability of state-of-the-art assistive devices for addressing the independent outdoor mobility needs of the visually impaired in an unstructured environment. We have created realistic datasets for various scenarios and evaluate deep neural networks for object detection on these datasets. We also present a portable prototype for the task. Further, we have also developed a cloud based solution to address the mobility requirements. We compare the local device based and cloud based solutions in terms of accuracy, latency, and energy. We present and discuss results from these two implementations that can provide insights for an effective solution. The results and insights open up novel research problems for embedded systems.**

I. Introduction

World Health Organization (WHO) report [1] states that a majority of visually impaired people reside in developing countries and survive in low income settings. Impaired vision (or complete blindness) creates additional hindrances for such people in performing their daily chores and limit their development and inclusion in the society. Being able to independently walk from one place to another (e.g., from residence to place of work or school) is essential for social and economic development of such differently abled population.

Unstructured infrastructure in developing countries pose significant challenges towards independent mobility of visually impaired. Various limitations of the solutions for developed country settings are already studied [2], [3]. Elmannai and Elleithy [4] present a survey of devices addressing mobility challenges. A large number of such devices are targeted at indoor mobility or require changes in infrastructure (e.g., bluetooth beacons). Our previous work [3] proposed MAVI (Mobility Assistant for Visually Impaired) to solve unique challenges for mobility in unstructured settings of developing countries. However, it did not consider feasibility of state-of-the-art solutions (e.g., neural network) for the object detection.

With the proliferation of network connectivity, using cloud services for various computer vision tasks have become feasible and attractive. In this work, we build a prototype of MAVI based on the cloud services and compare it to a local device based implementation. We present the effect of varying network bandwidth on different parameters of interest. Other works on cloud based assistive devices [5] do not address the challenges specific to the developing countries context. We identify and implement three primary functionalities in this prototype: Animal Detection (safety), Face Recognition (social inclusion) and Signboard Detection (navigation assistance).

Data is an indispensable requirement of modern statistical artificial intelligence, and novel deep learning models are as good as the data they have been trained on. We note that the public benchmark datasets available for object detection task do contain the classes of our interest (e.g., cow and dog). But, they are highly curated and often do not depict the real settings of an unstructured surroundings. Therefore, we have created and publicly release a new dataset [6] to address the specific needs of such a vision based mobility assistance system.

Though this paper focuses on analysis of different techniques, systems like MAVI must satisfy complex requirements with associated trade-offs on metrics such as accuracy, energy, etc. and adapt to variations in external factors (context) like walking speed, ambient lighting, etc. Hence, MAVI falls under a broader class of systems which we call Context-aware Adaptive Embedded Systems (CAES).

In summary, we claim the following contributions of this paper compared to our previous work [3] which provides further insights on ability of porting state-of-the-art computer vision techniques to embedded devices.

1) Release a dataset for animal detection and signboard detection to capture realistic scenarios and evaluate state-of-the-art machine learning/vision techniques on the same.
2) Implement various deep neural network (DNN) based tasks on embedded devices, using CPU only, and with acceleration. We analyze their comparative performance.
3) Implement a cloud services based prototype and analyze the performance over different network capabilities.

The rest of the paper is organized as follows. Section II gives a short overview of the MAVI. Section III provides details of dataset while Section IV explains implementation details. Section V gives a summary of the working prototype. Section VI presents various results which are analyzed in Section VII. Section VIII presents the summary and future directions.

II. MAVI Overview and Limitations of Prior Work

In this section, we present a quick review of the specification of the MAVI device, detailed in our previous work [3].

[1] Both the authors contributed equally to this paper. Contact Email: {kedia, anupamsobti}@cse.iitd.ac.in. MAVI website: http://www.cse.iitd.ac.in/mavi

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00058

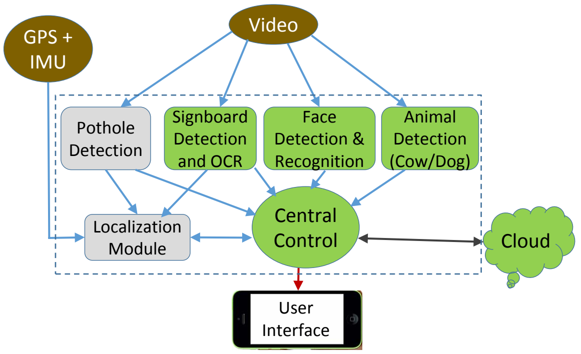


Fig. 1. Block diagram of MAVI. Blocks in green are the focus of this paper.

A. *Overview of MAVI device*

Fig. 1 shows a block diagram of MAVI which attempts to solve unique challenges in independent mobility of visually impaired pedestrians in unstructured environments. It implements four major computer vision tasks namely signboard detection (SBD) and optical character recognition (OCR), animal detection (AD), face detection (FD) and face recognition (FR), and pothole detection (PD). The device interfaces to the user through a mobile app communicating over a Bluetooth connection. An optional network interface can connect to the internet for cloud services to either retrieve a stored database for a specific task or to offload computations.

B. *Limitations of prior work*

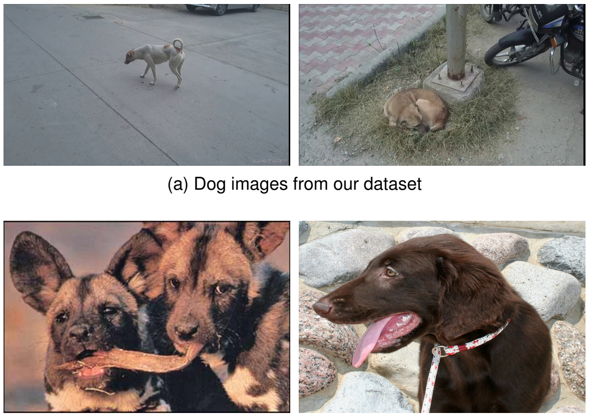
Section I already highlighted the key limitations of existing works. To the best of our knowledge, this is the first work which includes state-of-the-art methods for AD, FR and SBD with OCR on an embedded low-power portable platform.

The SBD module in our previous prototype [3] did not suggest any OCR implementation. The mobility aid for visually impaired by Poggi et al. [7] to detect obstacles using DNN does not consider stray animals as a valid detection class. Moreover, they do not address social inclusion as FR is not supported in their device. The prototype by Mocanu et al. [8] does not perform FR, AD, or SBD, limiting its applicability. Low latency AD is critical to the safety of the user, but is not implemented in any of the mobility aids.

With the improvement in network connectivity and cloud infrastructure, cloud based solutions are increasingly being deployed for computer vision tasks. Cloud based assistive device for visually impaired by Lapyko et al. [5] is limited in the features considered and does not compare its performance against a local implementation.

Further, there are no public datasets available for capturing the situations addressed by a system like MAVI. This limits any quantitative analysis of such systems. We believe that such a dataset will attract interest from computer vision researchers to improve the accuracy and runtime for such applications.

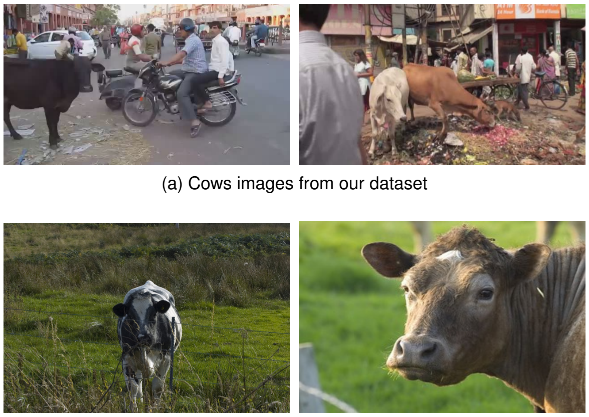
This paper details our approach to solving these limitations of existing mobility aids. We start with the dataset developed for MAVI, which we are releasing publicly with this work.



(a) Dog images from our dataset

(b) Dog images from Imagenet

Fig. 2. Example of images from our dog dataset compared to Imagenet.



(a) Cows images from our dataset

(b) Cows images from Imagenet

Fig. 3. Sample images from our cow dataset compared to Imagenet

III. Datasets for MAVI

Datasets are crucial to the successful implementation of the object detection techniques specially with the state-of-the-art approaches being data driven. Dog and cow images present in COCO [9] dataset, Imagenet [10], etc. do not make a good representation of the unstructured environments as shown in the Figures 2 and 3. Our evaluation of the performance of pre-trained object detection networks on our datasets, as discussed in Section VI-A2, also shows the scope of improvement, especially when used with small sized networks. Our dataset contains a huge variety of dogs and cows in sitting, standing, oblique and back views. For the signboard dataset, images from different angles and lighting conditions for the same signboard have been collected. Detailed statistics and characteristics of these datasets are shown in Table I. The dataset has been publicly released for further developments by the community [6].

IV. Implementation Details

We discuss both the offline (local implementation) and online (cloud based) versions of MAVI system in this section. It is important to note that a completely online system is not feasible due to limited internet connectivity in most rural areas

TABLE I
QUANTITY AND VARIETY IN MAVI DATASETS

Object Name	Number of Images	Annotations
Signboard	1493	Bounding Box, Text, Type, Direction, Conditions
Dog	1498	Bounding Box, Pose
Cow	1604	Bounding Box, Color, Pose, Occlusion

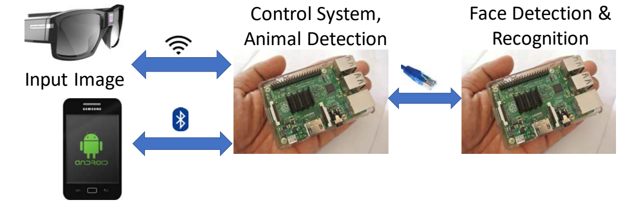


Fig. 4. Block diagram for local implementation of MAVI.

and intermittent connectivity, even in urban areas. Our online implementation uses off-the-shelf solutions from Google's Cloud APIs to implement different tasks in MAVI.

A. Local implementation

This section describes implementation of different tasks for an offline MAVI system as shown in Fig. 4. It uses a wireless transmitting camera (Pivothead [11]), two Raspberry Pi 3B [12] (R-Pi) and a power bank. The end result is spoken out via a mobile phone app. The camera API currently supports only RTSP (Real Time Streaming Protocol) encoding which needs decoding using an ffmpeg thread. Exploration of other cameras with lower processing cost is in process.

1) Face detection and recognition: In the last few years, there has been tremendous progress in the accuracy of face detection and recognition tasks using deep learning techniques. We have used OpenFace [13] framework for enabling face recognition in our prototype. The framework has a unique pipeline for processing faces and the cross-platform support for its dependencies has enabled us to use the same on R-Pi. We trained our network on 10 subjects with 15 images of each subject captured at various distances from the camera and poses. The training dataset size is in accordance with the OpenFace [13] framework. Due to optimized inference times on CPU provided by the library, we were able to run inference in ∼7s. A detailed analysis of time and energy are discussed in Section VI.

2) Animal detection: For the purpose of animal detection (cows and dogs), various pre-trained networks (listed in Table III) were analyzed. After an analysis of accuracy on our collected dataset, we selected SSD Mobilenet [14] model to perform rest of the experiments. In addition, we also accelerated the inference using a Movidius Neural Compute Stick (NCS) [15] (an accelerator for neural networks) for a more responsive animal detection module.

3) Signboard detection and OCR: There were two main challenges in signboard detection: text recognition (OCR) and detecting arrow symbols (for the navigation direction). We used a commonly used software named Tesseract [16] for OCR but inserted a number of custom pre-processing and post-processing steps to improve the accuracy. We used a polygon fitting algorithm to fit the detected signboard which enables us to remove any warping in the signboard text using a perspective transform. The next step, binarization, was implemented based on the work of Kasar et al. [17]. The binarized image is finally upscaled and given to the Tesseract OCR engine where the text prediction is done. The flowchart for the same is presented in Fig. 5. The output of the prediction engine is then post-processed by correlating with the words in the local dictionary. The local dictionary was built using all the words from the signboards in the dataset. The testing was done for bilingual signboards – English and Hindi. The signboard detection module is not yet integrated into the complete system owing to its slower speed, but works standalone on a R-Pi.

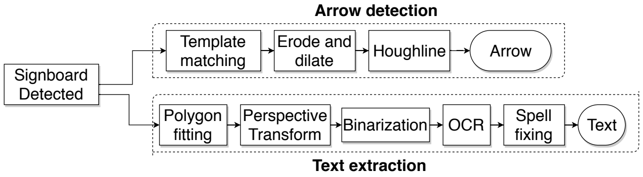


Fig. 5. Flowchart for signboard detection

TABLE II
FEATURES SUPPORTED BY DIFFERENT CLOUD SERVICE PROVIDERS

Service provider	FD	FR	AD	OCR
Google	Yes	No	Yes	Yes(EN,HI)
Microsoft	Yes	No	Yes	Yes(EN)
Amazon	Yes	Yes	Yes	Yes(EN)
IBM	Yes	No	Yes	Yes(EN)
SkyBiometry	Yes	Yes	No	No

1: EN refers to English OCR and HI refers to Hindi OCR

B. Cloud services based implementation

Cloud technology has shown considerable promise for offloading large computations from embedded devices, motivating us to explore it for MAVI. Our scope for exploring cloud based solution was limited to using the available solutions from cloud service providers. The main focus was on the analysis under different network conditions and ultimately looking for benefits and viability in comparison to local device based implementation. Firstly, we performed a comparison of computer vision features supported by leading cloud service providers, as shown in Table II. We find that many of the MAVI features like pothole detection and signboard detection are not supported by any of these service providers. Google Cloud was chosen as it was the best fit among the choices.

Google Cloud provides support for FD, text recognition (OCR) (can read characters, but not arrows), and AD (it can label the presence of animals, but cannot identify their position). Our implementation uses R-Pi as the platform which connects to the internet through WiFi or mobile data tethered using USB or WiFi hotspot. A python script invokes the Google Cloud vision API to send the image to cloud and receive the response containing the detection results. This response is parsed and communicated to the user through a mobile application, connected over Bluetooth. We use images

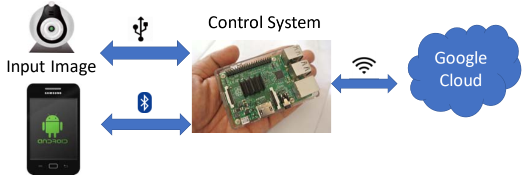


Fig. 6. Block diagram of MAVI on cloud

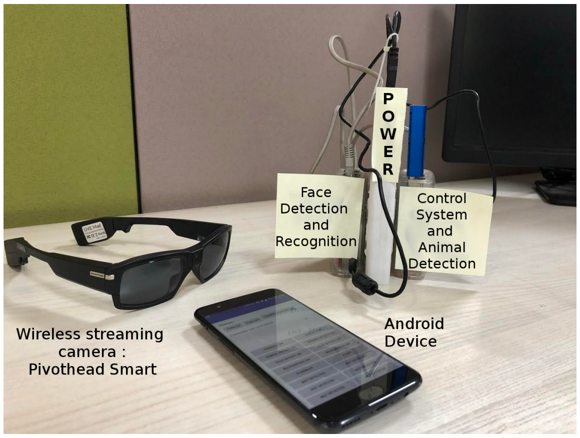

POWER
Face Detection and Recognition
Control System and Animal Detection
Android Device
Wireless streaming camera : Pivothead Smart

Fig. 7. A photograph of the complete prototype

from our dataset (refer Section III) stored on SD card of the R-Pi to perform measurements of accuracy, latency and energy. However, we use live feed from a USB camera connected to the R-Pi for the prototype system. A block diagram of this system is shown in Fig. 6.

V. Current Prototype

We describe the local and cloud based prototypes of MAVI in this section.

A. Prototype for the local implementation

The head-mounted spectacles (Pivothead) stream the imagery via WiFi. The two R-Pis connected via LAN and powered by the power bank can be carried in a bag. The output is sent to the phone where it is spoken out through the phone speaker or bluetooth headphones. This enables a hands-free solution to the user providing voice alerts as and when needed. The integrated system is shown in Fig. 7.

Control system and animal detection implementation: The control system is a multi-threaded process which incorporates the following functionality:

- **Frame Capture**: This thread captures the RTSP stream transmitted from Pivothead camera, decodes it using ffmpeg decoder, and then stores the stream as an image.
- **Frame Transmitter**: This thread transmits the captured image to the other R-Pi over a TCP socket and receives the FD and FR results from the same.
- **Animal Detection**: The AD thread initializes the NCS and sends every image to it. The received result is then parsed to extract the results of cow and dog detection.
- **Mobile Phone Transaction**: This thread sends the updated variables to the mobile application every 0.2 sec.

TABLE III
Animal Detection results on pre-trained models

Model Name	Dog Dataset mAP@0.5IOU	Cow Dataset mAP@0.5IOU
ssd_mobilenet_v1	0.47	0.65
ssd_inception_v2	0.56	0.72
rfcn_resnet	0.50	0.68
faster_rcnn_resnet	0.57	0.71
faster_rcnn_inception	0.67	0.78

Face recognition: The face recognition system, on a separate R-Pi, receives the image over LAN and sends the detection/recognition results back to the control system.

B. Prototype for cloud based implementation

The cloud implementation uses Google Cloud APIs to implement the task at hand. The detection results obtained from these APIs are communicated through the mobile application. This system does not use the Pivothead camera since WiFi interface of R-Pi is used to access Internet. We explored connecting Pivothead, R-Pi, and Internet on the same WiFi hotspot; which increased the image upload time, and thus the latency of tasks drastically. Hence, we used a USB camera.

VI. Results

We present the accuracy, runtime, and energy measured for different tasks for the local and cloud based implementation of MAVI.

A. Local implementation

1) Face detection and recognition: For a visually impaired person, we estimated the number of people to be kept in the database for recognition ability to be ∼50. Therefore, we evaluated the FR accuracy on a subset of standard LFW dataset containing 57 subjects with 25-50 images each. The accuracy was obtained as 97.59%. This proves OpenFace [13] as a viable candidate for our application in terms of accuracy. On R-Pi, we are able to run FD in 2.59s per image (VGA), with an additional overhead of 4.29s for FR per face. It uses ∼160MB of additional memory. This implementation is just the starting point and needs further improvement using quantization and DNN compression techniques to attain a faster processing.

2) Animal detection: We provide standard mAP [18] scores (a commonly used measure of accuracy) for different models on our datasets in Table III. However, as qualitatively observed in practice and also shown by Sobti et al. [19], a more responsive (and moderately accurate) detector provides better results on real-time feeds. Therefore, we use SSD Mobilenet and also accelerate it using Movidius NCS. The speed up is $\sim 4\times$. After fine-tuning ssd_mobilenet_v1 model for detection of cows and dogs, we were able to achieve a mAP of 0.73 in both categories, with 0.75 mAP for cows and 0.71 mAP for dogs. These numbers are on the test set for cows/dogs which were not present in the validation set (generated by a 20% split). The standalone runtime for AD is 1.2s on CPU and 0.3s when accelerated using NCS. The energy consumption is 2.1 mWh and 0.2 mWh for CPU and NCS respectively. Using NCS reduces the energy consumption by a factor of 10 and

TABLE IV
OCR ACCURACY RESULTS (IN %): COMPLETE, WITHOUT DICTIONARY (W/O D), AND WITHOUT PERSPECTIVE (W/O P) TRANSFORM. SOME IMAGES HAD SPECIFIC EFFECTS LIKE SKEW, GLARE, SHADOW AND BLUR WHICH ARE SEPARATELY PRESENTED TO HIGHLIGHT THE COMPLEXITY.

Type	Images	Complete		w/o D		w/o P	
		Eng.	Hin.	Eng.	Hin.	Eng.	Hin.
Overall	1493	70.18	52.44	47.85	30.19	60.04	43.81
Skew	165	69.28	53.14	48.15	29.92	57.32	40.85
Glare	41	34.42	22.56	25.48	13.84	29.31	18.31
Shadow	162	71.22	53.83	48.94	31.54	65.61	47.42
Blur	27	23.96	13.35	20.61	11.30	17.30	08.01

the algorithm can also run faster, thereby providing a better accuracy as per Sobti et al. [19]. However, this poses a tradeoff in terms of an additional component (cost, form factor, etc.) which needs to be analyzed to make decisions.

3) Signboard detection and OCR: Table IV shows the effect of different pre/post-processing steps which are applied for signboard detection and OCR subsystem. Perspective transform alone improves the accuracy by over 10%, while inclusion of a dictionary improves the accuracy by over 20%. This shows the necessity of these transformations. The table also show the reduced accuracy in certain image conditions, indicating the need of algorithmic interventions required to improve the performance further. Our implementation takes an average of 12.54s on the R-Pi with an energy consumption of 4.4 mWh per image. The implementation needs to be accelerated to reduce the runtime.

B. Cloud based implementation

We present the results obtained from the cloud based implementation of MAVI. We plot the latency for uploading image to cloud and the total end-to-end time for processing each image for different network connections in Fig. 8. The latency follows the order 3G >4G >WiFi, owing to their increasing order in terms of supported bandwidth. Moreover, we also observe higher variance for lower bandwidth network. Fig. 9 shows the variations in the cloud runtime for different tasks. One could see that for the OCR task, using dual language OCR mode (English + Hindi) takes significantly more time than using English-only mode and has a larger variance too.

The energy consumption to process images using cloud services was also measured for a batch of images sent to cloud together and corresponding detection results received. We obtain an energy consumption of 0.974 mWh per image and an average current consumption of 361 mA when internet is tethered using WiFi hotspot. We obtain an energy reading of 2.809 mWh and average current consumption of 824.5 mA when internet is tethered using USB. The reason for increase in current for USB is that the phone draws about 500mA current for charging itself, when connected over USB. We could not turn off the charging and thus conclude WiFi tethering to be a better choice than USB to reduce energy consumption.

The accuracy results for MAVI for different tasks running on cloud is captured in Table V. For tasks like AD and FD, the obtained accuracy is lower on cloud compared to the ones implemented by us on the local device. This indicates that MAVI specifications pose a challenging detection problem even for the state-of-the-art cloud systems.

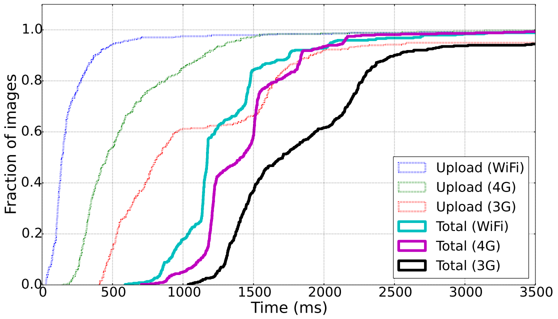


Fig. 8. CDF for image upload time and total time for processing different tasks: We plot a cumulative distribution function (CDF) to indicate the fraction of images (y-axis) taking latency less than a given time (x-axis). Upload time and total time for 3G>4G>WiFi

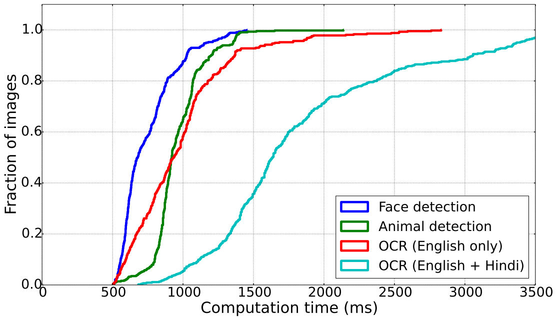


Fig. 9. CDF for computation time for different algorithms on cloud: Bilingual OCR takes much longer time than English OCR and has more variance as well.

VII. COMPARISON OF LOCAL AND CLOUD BASED IMPLEMENTATION

In this section, the results obtained from local and cloud based implementations of MAVI tasks are compared. Subsequently, the limitations of each implementation is presented, making a case for a hybrid solution to address the target application.

Differences in implementation: It is important to note that the algorithms used for the two implementations are very different from each other and are often unknown for the cloud. AD is implemented locally as an object detection task which returns the bounding box (position of animal) as well. However, the cloud solution merely returns the presence/absence of animals. Similarly, FR is not supported on the cloud. For SBD, cloud supports OCR which can detect text but not arrows (for directions). Despite these differences, it is useful to compare them to get a sense of what can be achieved through state-of-the-art cloud services vs. custom implementations.

A. Latency

The latency obtained using cloud services depends upon the available network bandwidth and ranges from 1s at high

TABLE V
ACCURACY (%) FOR DIFFERENT ALGORITHMS ON CLOUD

AD (cow)	AD(dog)	FD	OCR(Eng.)	OCR(Hin.)
69.18	51.92	55.27	90.26	60.58

bandwidth to 3s for low bandwidths (the bandwidth would be further lower for 2G and hence higher latency). However, the average latency for AD being run locally is 1.2s on CPU and 0.3s when accelerated using NCS. FD takes $\sim$ 2.6s on average to run locally and FR takes additional 4.3s. On cloud, FD takes 1s to 2s while FR is not supported.

B. Accuracy

We expected cloud based implementation to provide a superior accuracy due to the usage of larger resources and state-of-the-art algorithms. However, the results obtained for accuracy indicate a mixed behavior between local and cloud implementation. For AD, we observe a low accuracy on cloud as well as the local device based implementation. One key reason for this is the presence of the dataset targeting realistic scenarios, which pose significant challenges to the existing algorithms. This also depicts that the problem being solved by this work is indeed challenging and needs a collaborative effort from researchers in computer vision and embedded domain. When re-trained and tuned, the local implementation for AD shows improved accuracy. This is not possible for cloud solution. FD on local system achieves much higher accuracy than the cloud solution which misses out small sized faces in the image.

C. Energy

Energy consumption is a critical parameter for MAVI as it is a portable and battery powered device. The cloud based solution is highly energy efficient as compared to local processing. However, we do not account for the significant energy which gets consumed to maintain the cloud infrastructure.

Overall, we see that cloud based solution can provide a better runtime and energy consumption than local implementation when the network signal and bandwidth is good. However, there are limitations on the features supported by cloud and also variability due to bandwidth. Moreover, the cost of using the cloud services and the network is currently ignored, which may play an important role in decision making. The results indicate that an optimal implementation should use a hybrid solution where available services are used from cloud and other algorithms are implemented locally. Moreover, for tasks which could run on either cloud or local, it should be possible to switch between them depending upon the available bandwidth. Such a switching based on network bandwidth is an example of adaptive aspect for a CAES like MAVI. We expect a hybrid solution to bring-up some interesting challenges in implementation and new insights to emerge from this exercise that we leave as a future work at this moment.

VIII. CONCLUSION AND FUTURE WORK

We presented a local device based and a cloud implementation of MAVI system to aid mobility of visually impaired. The presented implementation uses state-of-the-art techniques and quantifies the tradeoff w.r.t. a cloud based solution in terms of accuracy, latency, and energy usage. The analysis motivates the case for a hybrid solution of online/offline processing and requires development/usage of smart scheduling algorithms for a responsive system. It demonstrates the use of deep neural networks (DNNs) and DNN accelerator hardware in a portable mobility assistance prototype.

As part of the future work, an integrated cloud and local solution would be able to harness the benefits of both. The face recognition (FR) could also be accelerated on NCS and further improvements in runtime could be obtained using quantization of DNN, with a reasonable tradeoff in accuracy. We are also exploring improvement in accuracy of various tasks with the help of additional sensors.

ACKNOWLEDGEMENTS

The authors would like to thank MeitY for funding MAVI under the project SMDP-C2SD and providing assistantship to Rajesh Kedia and Anupam Sobti under Visvesvaraya PhD fellowship programme. We also acknowledge efforts of all students who contributed to MAVI in previous years.

REFERENCES

[1] "World health organization. visual impairment and blindness." www.who.int/mediacentre/factsheets/fs282/en.

[2] P. Chanana, R. Paul, M. Balakrishnan, and P. Rao, "Assistive technology solutions for aiding travel of pedestrians with visual impairment," *J. of Rehabilitation and Assistive Technologies Engineering*, vol. 4, 2017.

[3] R. Kedia, K. K. Yoosuf, P. Dedeepya, M. Fazal, C. Arora, and M. Balakrishnan, "MAVI: An embedded device to assist mobility of visually impaired," in *VLSID*. IEEE, 2017.

[4] W. Elmannai and K. Elleithy, "Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions," *Sensors*, vol. 17, no. 3, 2017.

[5] A. N. Lapyko, L. P. Tung, and B. S. P. Lin, "A cloud-based outdoor assistive navigation system for the blind and visually impaired," in *7th IFIP Wireless and Mobile Networking Conference (WMNC)*, May 2014.

[6] "Datasets for MAVI," http://www.cse.iitd.ac.in/mavi/datasets.html.

[7] M. Poggi and S. Mattoccia, "A wearable mobility aid for the visually impaired based on embedded 3D vision and deep learning," in *IEEE Symposium on Computers and Communication (ISCC)*, 2016.

[8] B. Mocanu, R. Tapu, and T. Zaharia, "When ultrasonic sensors and computer vision join forces for efficient obstacle detection and recognition," *Sensors*, vol. 16, no. 11, 2016.

[9] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*. Springer, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012.

[11] "Pivothead," http://www.pivothead.com/smart-2.

[12] "Raspberry pi 3 model b," https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[13] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[14] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint 1704.04861*, 2017.

[15] "Movidius," https://developer.movidius.com.

[16] "Tesseract," https://github.com/tesseract-ocr.

[17] T. Kasar, J. Kumar, and A. Ramakrishnan, "Font and background color independent text binarization," in *Second international workshop on camera-based document analysis and recognition*, 2007.

[18] M. Everingham *et al.*, "The Pascal visual object classes (VOC) challenge," *International journal of computer vision*, vol. 88, no. 2, 2010.

[19] A. Sobti, C. Arora, and M. Balakrishnan, "Object detection in real-time systems: Going beyond precision," in *WACV*. IEEE, 2018.