Computer Vision-enhanced Augmented Reality for mountain outdoor exploration

Darian Frajberg, Piero Fraternali
Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria,
Piazza Leonardo da Vinci, 32, Milan, Italy
{first.last}@polimi.it

Abstract

Outdoor Augmented Reality applications are an emerging class of software systems that demand the fast identification of natural objects on mobile and embedded systems. They arise as an interesting tool to support the creation of entertainment applications, which can be further boosted by the introduction of Deep Learning-based Computer Vision algorithms. Nonetheless, their execution remains challenging and requires non negligible resources for devices with hardware constraints. This paper discusses the experience of a real world mobile Augmented Reality application for mountain exploration.

1. Introduction

Outdoor mobile Augmented Reality (AR) applications promote a new way of marketing the touristic offer of a territory, by overlaying useful information directly on top of the user's camera view. These applications aim at exploiting the sensor readings (GPS position, phone orientation and possibly also the camera) of the device to understand what the user is looking at and enrich the view with contextual information to enable knowledge acquisition and exploration. Furthermore, Deep Learning (DL) has recently exhibited superior performance in a wide variety of Computer Vision (CV) tasks and can lead to novel AR solutions.

The development of outdoor mobile AR apps poses several technical challenges: 1) they must understand the current context and activity of the user; 2) they must find the appropriate information pertinent to the user's current activity and view; 3) they must overlay the retrieved information onto the device screen in a way that is adequate to the user's experience. Besides, also non functional requirements are relevant: the application must work in real-time on devices with limited computational power and strict energy consumption constraints, and functioning should ideally be guaranteed also in absence of Internet connectivity. Taken together, all such requirements make the development of outdoor mobile AR apps a non trivial task.

Figure 1. PeakLens app.

This paper discusses the creation and support of applications of this nature by presenting the experience of PeakLens[1] (Figure 1), a real world mobile app that combines AR and CV for the identification of mountain peaks; PeakLens processes sensor readings and camera frames in real-time by using an efficient on-board DL-powered CV module. It is available for Android and has over 400k installs.

2. Related Work

Augmented Reality applications.
AR is a well established research area in the Human-Computer Interaction field, which has recently gained momentum due to the introduction by major hardware vendors of consumer-grade AR wearable devices (e.g. Microsoft HoloLens, Magic Leap One and Google glasses). Furthermore, a recent trend shows mobile devices being used as low cost AR platforms without requiring ad-hoc hardware [9]. An important branch of the discipline is the development of outdoor AR mobile apps to identify and

[1] https://www.peaklens.com

track points of interest in urban or rural scenarios [10][13]. However, they generally rely on GPS and orientation sensors or on the specific a priori known appearance of certain objects, without actually considering the frequent sensor readings' noise, the non-stationary nature of outdoor environments, and the value and complexity of the visual content captured by the camera of the device.

Efficient mobile Deep Learning.
Artificial Intelligence (AI) on the edge is a matter of great importance towards the enhancement of smart devices that rely on hardware with limited computational power and operations with real-time constraints [11]. In particular, DL has the potential to achieve remarkable results, but the deployment of big and complex models is not completely suitable for embedded systems, such as smartphones, tablets and wearable devices. Several works address compression, acceleration and definition of lightweight architectures for DL models expressly conceived for efficient deployment [7][15][8]. Their embedding within AR apps is of particular interest, as well as the profiling, benchmarking and optimization of mobile DL frameworks.

Environmental citizen science.
Environment data collection more and more exploits the contribution of citizens, who cooperate through their mobile phones for acquiring and processing large geo-referenced datasets and for extracting from them information usable in the study of natural and anthropic processes. The main challenge of developing crowdsourcing applications for large scale environment geo-data collection is to offer citizens a useful and possibly entertaining experience, so as to motivate them to participate [6][14]. As a use case, this work targets the environmental monitoring of mountains [1].

3. Image-based geo-localization

Several studies have proposed the use of image-based techniques for large scale visual geo-localization purposes [12]. In particular, image-to-terrain techniques are suitable for both geo-localization and camera orientation estimation, when dealing with images taken in natural mountain environments. Such problem can be tackled by computing the alignment between the skylines extracted from Digital Elevation Model (DEM) data and mountain images. Heuristic methods based on edge detection work well on images taken in good conditions, but present difficulties with bad weather or occluded scenarios. The work in [4] presents the results of training a Fully Convolutional Neural Network for the fast and accurate extraction of mountain skylines. It was trained at patch-level for binary classification and can be executed for pixel-wise classification over full images. For training and evaluation, we manually annotated the skyline of $\approx$ 9,000 mountain images fetched from Flickr and from over 2,000 touristic web-cams. Images in the data set are complex, diverse and contain a variety of obstacles occluding the skyline. Overall evaluation comprised the definition of metric functions that assess image level quality by comparing the skyline extracted by the CNN model with the one manually annotated in the ground truth. This work has been further extended by improving accuracy and incorporating lightweight layers [8] to optimize deployment.

4. CV-based outdoor mobile AR

The distinctive characteristic of mobile AR applications is the overlay of information directly on top of what the user sees, based on the user's context estimated from the device sensors. Nonetheless, such sensors are often noisy and, for many AR applications, small errors on the position or orientation estimation (i.e. 5 degrees) can totally ruin the user experience. Fortunately, outdoor mobile AR applications can highly benefit from CV solutions able to exploit the visual content captured by the camera in order to compensate existing noisy sensor readings. As a result, computed final projections are enhanced and the user experience is improved. The work in [2] discusses the implementation of PeakLens, a real world outdoor mobile AR app for mountain peak identification, which has the potential to be used to crowdsource mountain images for environmental purposes, such as the analysis and monitoring of snow coverage for water availability prediction. PeakLens identifies mountain peaks and overlays them in real-time on the camera view. Besides projections based on noisy mobile orientation sensors, it features a CV module based on DL (running on-board the phone) for the extraction of the mountain skyline from the camera frames; such skyline is subsequently aligned w.r.t. a virtual panorama computed from the GPS position of the user and a DEM of the Earth. As a result, GPS and compass errors are corrected and the peak projections are significantly improved.

5. Optimized DL inference for mobile systems

Despite the rapid growth of computational power in embedded systems, the deployment of highly complex and considerably big models remains challenging. Optimized execution requires managing memory allocation efficiently, to avoid overloading, and exploiting the available hardware resources for acceleration, which is not trivial given the non standardized access to such resources. When dealing with specific hardware architectures or specific silicon vendors, maximum performance can be reached by developing ad-hoc optimized solutions. Nonetheless, such approach may comprise scalability and maintenance issues when targeting a high number of extremely heterogeneous architectures and devices, as in the case of Android market nowadays. We have built PolimiDL[2] [3], an open source, publicly available framework for accelerating DL inference on mobile devices and embedded systems, when no off-the-shelf solutions for the deployment on such devices were available. PolimiDL does not focus on training, but on speeding-up the execution-time of ready-to-use models by applying multiple optimization methods and increases efficiency of operations without impacting accuracy. Its implementation is very generic, with neither hardware nor platform specific components, and supports devices with very heterogeneous architectures. Experiments have shown highly competitive results with respect to TensorFlow Lite for the deployment of small models on a set of heterogeneous mobile devices. In particular, its average inference time for the optimized skyline extraction model of PeakLens was reduced by above 60% when confronted with TensorFlow Lite.

6. Testing Multi-Sensor Mobile Applications

Outdoor mobile applications rely on the input of multiple, possibly noisy sensors, such as camera, GPS, magnetometer, accelerometer, gyroscope and microphone. Testing such applications requires the reproduction of the real conditions in which the application works, which are hard to recreate without automated support. This is particularly challenging when CV is used, due to the necessity of recreating the relationship between camera frames and other sensor readings that help infer orientation, motion and view. The work in [5] tackles such problem by proposing a Capture & Replay framework that automates the testing of outdoor mobile applications; the framework records in real-time data streams from multiple sensors acquired in field conditions, stores them, and let developers replay recorded test sequences in lab conditions, also computing quality metrics to measure performance and help tracing errors. This framework supports the development and testing of CV-based outdoor AR systems depending on multiple correlated sensor streams, which is the case of PeakLens app.

7. Conclusions

We have illustrated the ongoing research, which pursues the combination of CV and AR for the creation of outdoor mobile applications for entertainment and environment data collection. Future work will focus on further optimizing DL for embedded systems and testing it in real applications. Moreover, Artificial Intelligence techniques will be exploited for knowledge extraction and for the aggregation of such derived information into environmental models.

References

[1] A. Castelletti, R. Fedorov, P. Fraternali, and M. Giuliani. Multimedia on the mountaintop: Using public snow images to improve water systems operation. In *24th International Conference on Multimedia*. ACM, 2016.

[2] R. Fedorov, D. Frajberg, and P. Fraternali. A framework for outdoor mobile augmented reality and its application to mountain peak detection. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*. Springer, 2016.

[3] D. Frajberg, C. Bernaschina, C. Marone, and P. Fraternali. Accelerating deep learning inference on mobile systems. In *International Conference on AI and Mobile Services*. Springer, 2019.

[4] D. Frajberg, P. Fraternali, and R. N. Torres. Convolutional neural network for pixel-wise skyline detection. In *International Conference on Artificial Neural Networks*. Springer, 2017.

[5] D. Frajberg, P. Fraternali, R. N. Torres, C. Bernaschina, and R. Fedorov. A testing framework for multi-sensor mobile applications. *Journal of Mobile Multimedia*, 15(1):1–28, 2019.

[6] H. Goëau, P. Bonnet, A. Joly, V. Bakić, J. Barbe, I. Yahiaoui, S. Selmi, J. Carré, D. Barthélémy, N. Boujemaa, et al. Pl@ntnet mobile app. In *Proceedings of the 21st International Conference on Multimedia*. ACM, 2013.

[7] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, 2015.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] P. Jain, J. Manweiler, and R. Roy Choudhury. Overlay: Practical mobile augmented reality. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2015.

[10] P. Kourouthanassis, C. Boletsis, C. Bardaki, and D. Chasanidou. Tourists responses to mobile augmented reality travel guides: The role of emotions on adoption behavior. *Pervasive and Mobile Computing*, 2015.

[11] N. D. Lane and P. Georgiev. Can deep learning revolutionize mobile sensing? In *International Workshop on Mobile Computing Systems and Applications*. ACM, 2015.

[12] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand. Geolocalization using skylines from omni-images. In *12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 2009.

[13] P. A. Rauschnabel, A. Rossmann, and M. C. tom Dieck. An adoption framework for mobile augmented reality games: The case of pokémon go. *Computers in Human Behavior*, 2017.

[14] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015.

[15] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016.

[2]https://github.com/darianfrajberg/polimidl