# Deep Neural Networks in embedded systems for counters

Anderson Tavares[1], Jens Lundström[2], Stefan Byttner[2] and Maycel Faraj[1]

***Abstract*— This paper discusses an embedded system prototype able to do privacy-preserving location analytics, a systems that does not store any personal information. More specifically, the functionality that is discussed in this paper relates to counting the number of people that are present in a predefined space using a deep neural network (DNN) algorithm. The algorithm is optimized for low computational complexity in order to be able to execute onboard an embedded system while retaining sufficiently high accuracy with robustness to variations in scale, viewpoint, and lighting conditions.**

## I. INTRODUCTION

Today, the company Synteda offers the product Be-Metrics for customers such as hotels, offices, convention centers, shopping centers and other large facilities that seek a deeper understanding of their customers' movement patterns by measuring mobile phone positions. Through the product that makes e.g. shopping centers become aware of the movement patterns of customers, Be-Metrics enables a multitude of opportunities to improve the customer experience and to customize marketing and campaigns. The core of the current service is a WiFi-based location equipment that calculates the position of the surrounding mobile phones according to the multilateration principle. The current solution uses a cost-effective hardware that can be scaled in number so that a good estimation of mobile positions can be achieved. The following digital services and values can be created:

- Identification of movement patterns: Individual digital paths are identified and clustered into well-visited paths that are presented automatically and easily understood by the user. The value lies in the fact that the user (Synteda's customer) can see how their shopping centre, exhibition hall, office or hotel is used by their visitors. Furthermore, architects or store planners can change a physical environment and study how visitor behavior changes (e.g. in terms of how customers visit shops or use café restaurants).
- Prediction of movement patterns: Through ML, the possibility opens up for the system to be able to predict at what times different office/shops/places/restaurants are visited but also the number of visits.

In many cities there are shopping malls built, large open-space building where people meet, eat and shop. As many cities get outlets on the outskirts of the cities the city center is depopulated and not rarely city centers become a *sleep town*. If people stay longer in the mall, they socialize which is a pattern interesting to detect in order to develop shopping malls and similar buildings. When you connect data from shopping centers in different cities, it should be possible to analyze and predict what should be done in the shopping center to attract visitors, retain them and create square-like environments that make people stay longer, meet and enjoy themselves. With Be-metrics, it becomes possible to find these patterns and draw conclusions about what can be done to make things even better.

Coupled with location analytics there is also the possibility to use additional sensors such as cameras. This makes it possible to count the number of people entering a defined space, independently of whether they are having a WIFI-device or not. This allows for a secondary information source for verification of the number of people in a particular space and location.

The challenge here is that the image processing is done in a local embedded node with deep learning algorithms typically both demanding in energy and computations. This paper presents a study on the considerations that were done in order to make a people-counting deep neural network algorithm feasible to run in such an embedded system and to preserve any personal information, while retaining as high accuracy as possible.

## II. STATE OF ART FOR DEEP LEARNING IN EMBEDDED SYSTEMS

There are numerous proposed methods and architectures for image-based person detection in embedded systems. Early research often adopts the use of manual feature extraction methods such as Haar features or the Histogram of Oriented Gradients (HOG) followed by a Support Vector Machine classifier or a shallow artificial neural networks [1], specifically targeting pedestrian detection. As embedded hardware decrease in cost and increase in performance, later research and examples utilize Convolution Neural Networks (CNN) in order to detect the bounding box of person of per-pixel classification. Among many CNN architectures the YOLO-v3 has been standing out in detection performance [5]. An example of such implementation is by Schrijvers et al [2] which adopts the YOLO-v3 CNN to detect persons in a shopping area in order to track and draw insights form local consumer behaviour. Schrijvers et al also mentions the necessity of having a lightweight network able to run in real-time at a cost-effective platform. These requirements together with the correlated requirement of having an energy efficient system has lead researchers to point to more efficient neural network architectures [3] such as DenseNet [6]. Other

*This work was supported by Synteda AB

[1]A. Tavares and M. Faraj is with Synteda AB. anderson at synteda.com and mi at synteda.com

[2]S. Byttner and J. Lundstrom is with Halmstad University. stefan.byttner at hh.se and jens.r.lundstrom at hh.se


Fig. 1. Example of people counter system.

examples include the reduction of connections in the already trained neural network (i.e. *network compression*), in [4] Fang et al reduced the popular YOLO-v3 to a compressed version, Tinier-YOLO able to perform real-time tasks in a more efficient way.

## III. System overview

The system use simple single-board computers in combination with sensor utilizing deep neural networks (DNN) combined to count number of heads crossing a space. The captured data is used only at the embedded system and deleted (without any caching) after calculation, to remove any personal information. The system is built on the four different stages: detection, matching, tracking and counting. Using the detection we created a data set to be able to capture many possible movements and abnormal behaviors. After training the DNN we performed tests to evaluate the performance of the algorithm, see an example of such detection of a head in Figure 1.

### A. Detection

The detection model receive as inputs images $I(w, h, c)$ either from a camera or a recorded video file, where $w$, $h$, and $c$ is image width, height and color information respectively. Then, the algorithm outputs a list of bounding boxes $b_i = [c_x, c_y, w, h]$ for each frame representing the position of each detected person ($c_x$ and $c_y$ denotes center pixel position of the bounding box). Each frame is independent from each other, therefore a matching step is necessary.

### B. Matching

In order to group bounding boxes from different frames that might represent the same person, we match detected bounding boxes and already tracked people. The matching process finds correlations between the predicted next position of the person from the tracker and the detected bounding box. It outputs a table of correlations, where we can extract the best bounding box for each person. We can also detect whether a given person was not found. Moreover, this process also finds bounding boxes with low correlation score with existing people, which is interpreted as a new person to be added to the list.

### C. Tracking

There is one tracker for each person in a sequence of frames. The tracker process has two steps:

- *prediction*, of where the position of the person is changed to based on its internal state (e.g., estimated velocity and acceleration);
- *update*, of where the prediction is corrected from real measurements (e.g., bounding boxes).

### D. Counting

The counting process takes the centroid $c_{i,n}$ of each person $i$ at frame $n$ and compares with a virtual line $l$ which represents a pass-through way. By using homogeneous coordinates, we can define which side the person is:

$$s_n = \text{sign}(\langle c_{i,n}, l \rangle) \tag{1}$$

By comparing the previous and the current states, if $s_n \neq s_{n-1}$, then the person has crossed the line and we update the counter (a person leaving or entering the area).

## IV. Conclusions

To meet several challenges of embedded system-based people counting we have in this paper described the utilization of methods such as deep learning techniques optimized for low computational complexity. The system has shown in our tests to be able to execute well within desired framerate and people counting accuracy, as well as to preserve any personal information. By utilizing the DNN we have an advantage, with a validated data set, to perform better to learn and adapt the system much faster by eliminating the need for hand-crafted features for each occurred situation or issue. The system execute onboard an embedded system shows to be more robust to variations in scale, viewpoint, and lighting conditions and retaining sufficiently high accuracy.

## References

[1] Brunetti, Antonio, et al. "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey." Neurocomputing 300 (2018): 17-33.

[2] Schrijvers, Robin, et al. "Real-time embedded person detection and tracking for shopping behaviour analysis." Advanced Concepts for Intelligent Vision Systems: 20th International Conference, ACIVS 2020, Auckland, New Zealand, February 10–14, 2020, Proceedings 20. Springer International Publishing, 2020.

[3] Chen, Yanjiao, et al. "Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions." ACM Computing Surveys (CSUR) 53.4 (2020): 1-37.

[4] Fang, Wei, Lin Wang, and Peiming Ren. "Tinier-YOLO: A real-time object detection method for constrained environments." IEEE Access 8 (2019): 1935-1944.

[5] Kim, Chloe Eunhyang, et al. "A comparison of embedded deep learning methods for person detection." arXiv preprint arXiv:1812.03451 (2018).

[6] Gao Huang et al., "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, doi: 10.1109/CVPR.2017.243.